Riichiro Mizoguchi
Zhongzhi Shi
Fausto Giunchiglia (Eds.)

LNCS 4185

The Semantic Web – ASWC 2006

First Asian Semantic Web Conference
Beijing, China, September 2006
Proceedings

Springer

Lecture Notes in Computer Science 4185

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

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

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

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

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

Bernhard Steffen
University of Dortmund, Germany

Madhu Sudan
Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos
University of California, Los Angeles, CA, USA

Doug Tygar
University of California, Berkeley, CA, USA

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

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

Riichiro Mizoguchi Zhongzhi Shi
Fausto Giunchiglia (Eds.)

The Semantic Web – ASWC 2006

First Asian Semantic Web Conference
Beijing, China, September 3-7, 2006
Proceedings

Springer

Volume Editors

Riichiro Mizoguchi
The Institute of Scientific and Industrial Research
Osaka University
Osaka, 567-0047 Japan
E-mail: miz@ei.sanken.osaka-u.ac.jp

Zhongzhi Shi
Institute of Computing Technology
Chinese Academy of Science
Beijing 100080, China
E-mail: shizz@ics.ict.ac.cn

Fausto Giunchiglia
Department of Information and Communication Technology
University of Trento, Italy
E-mail: fausto@dit.unitn.it

Library of Congress Control Number: 2006931395

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

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

ISSN 0302-9743
ISBN-10 3-540-38329-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-38329-1 Springer Berlin Heidelberg New York

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

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2006

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11836025 06/3142 5 4 3 2 1 0

Preface

The International Semantic Web Conference (ISWC) and the European Semantic Web Conference (ESWC) present the latest results in research and application of the Semantic Web technologies. Both have contributed to the promotion of research on the Semantic Web in their respective regions. Research on the Semantic Web needs global activity which necessarily requires the spread of the Semantic Web over Asia where it has been under development. The series of Asian Semantic Web Conferences (ASWC) have therefore been established with the intention of fostering research and development of the Semantic Web and its related technology in Asia by the East Web project, http://odle.dit.unitn.it/eastweb/, whose objectives include fostering and promoting the cooperation between European and Asian Institutions involved in IT education and research. The first ASWC was held in Beijing, during September 3–7, 2006, in this context.

We initially received 253 submissions and found 221 valid submissions of abstracts after a screening process. We finally received 208 full papers each of which was reviewed seriously by three Program Committee members and we accepted 36 full papers and 36 short papers. The acceptance rate of full papers is 18%, which we are proud of. The acceptance rate of all the accepted papers is 36%. Differently from ISWC/ESWC, industrial track papers of ASWC 2006 were reviewed by the Program Committee of the research track with the same quality level but with different criteria, that is, practicality was considered more important than originality. We accepted eight papers, four of them are full papers and four short papers, which are included in the above-mentioned 72 papers. The major characteristic of the topics of ASWC 2006 is that 1/4 of the total papers are ontology related. Topics covered by the accepted papers are as follows:

Ontology-related papers:		18
Ontology integration and interoperability	7	
Ontology alignment	4	
Ontology and theory	4	
Ontology and tools	3	
Applications		10
Semantic Web services		9
Reasoning		5
Annotation		4
Social network and RSS		4
Peer-to-Peer		4
Database		4
Information search		3
Document and recommendation		3
Industrial track		8

Accepted papers come from 18 countries, which shows that ASWC 2006 is quite international, and their statistics in terms of country are as follows:

China	30
Korea	11
Japan	10
Ireland	4
Austria	2
Finland	2
USA	2
Australia	1
Belgium	1
France	1
Germany	1
Greece	1
Iran	1
Italy	1
Kuwait	1
Norway	1
Thailand	1
UK	1

ASWC 2006 consisted of a three-day main conference which included paper and poster tracks and three invited talks, a two-day workshop/tutorial and an Industrial Day. The three invited speakers were Jim Hendler, University of Maryland at College Park, USA, Hai Zhgue, Institute of Computing Technology, Chinese Academy of Sciences, China and Enrico Motta, The Open University, UK.

Jim Hendler talked about KR issues in the Semantic Web era under the title of "The Semantic Web: A Network of Understanding." He discussed major characteristics of the new-generation KR such as "extra-logical" infrastructure, semantic interoperability beyond a syntactic one, heterogeneity, scalability and so on. It was also his intention to confirm that Semantic Web KR is different from traditional AI. Hai Zhgue's talk was entitled "Transformation from OWL Description to Resource Space Model." He discussed the necessity of the synergy of semantics in the real world, the document world and the mental abstraction world. On the basis of his resource space model (RSM), he discussed an automatic translation of OWL descriptions into resource space as a step toward his ultimate goal. The killer applications of the Semantic Web were one of the serious topics. Enrico Motta discussed the topic in his talk on "Next-Generation Semantic Web Applications." He analyzed the current state of the art of Semantic Web applications followed by their main features and stressed the importance of shifting from the first-generation to the second-generation applications by exploiting the increased heterogeneity of semantic sources.

Before the main conference, we had seven workshops:

- Making the Semantic Web Services Relevant to Industry
- Semantic e-science
- Semantic Web Education and Training
- Semantic Technologies, Educational Standards, e-Learning Application Vocabularies, and OpenCourseWare
- Semantic Web Applications and Tools Workshop
- Web Search Technology—from Search to Semantic Search
- Service Discovery on the WWW

and three tutorials:

- Semantic Web Services—State of Affairs
- XML Query Reformulation for XPath, XSLT and XQuery
- Tools and Applications for the Corporate Semantic Web

All the events arranged in ASWC 2006 were very successful and contributed to the facilitation of Semantic Web research in Asia as well as the cross-fertilization among researchers working in academia and industries. We believe we have made a good start to the ASWC series.

As Program Committee Co-chairs and Conference Chair, we are grateful to the Program Committee members listed below and to the additional reviewers for their enormous effort in reviewing to select these wonderful papers. Without their contribution, this conference would not have happened. Considering ASWC 2006 was the first conference in Asia, the organization went smoothly thanks to the strong leadership of the Local Organizing Committee Chair, Juanzi Li, to whom our special thanks go. We also would like to thank the sponsors listed below for their monetary support, which was another key factor of the great success of ASWC 2006.

Riichiro Mizoguchi
Program Committee Chair

Zhongzhi Shi
Local Co-chair

Fausto Giunchiglia
Conference Chair

Organizing Committee

Conference Chair:	Fausto Giunchiglia (University of Trento, Italy)
Local Conference Co-chairs:	Bo Zhang (Tsinghua University, China)
	Ruqian Lu (Chinese Academy of Science, China)
	Shiqiang Yang (Tsinghua University, China)
Program Committee Chair:	Riichiro Mizoguchi (Osaka University, Japan)
Local Co-chair:	Zhongzhi Shi (Chinese Academy of Science, China)
Local Organizing Chair:	Juanzi Li (Tsinghua University, China)
Tutorial Co-chairs:	Ying Ding (DERI, Austria)
	Hai Zhuge (Chinese Academy of Science, China)
	Maosong Sun (Tsinghua University, China)
Workshop Co-chairs:	Marco Ronchetti (University of Trento, Italy)
	Guohui Li (National University of Defense Technology, China)
Industrial Track Co-chairs:	Alain Leger (France Telecom, France)
	Vilas Wuwongse (Asian Institute of Technology, Thailand)
	Xin sheng Mao (IBM CSDL, China)
Demo Co-chairs:	Michal Zaremba (DERI, Austria)
	Guangwen Yang (Tsinghua University, China)
Sponsor Co-chairs:	York Sure (University of Karlsruhe, Germany)
	Bin Xu (Tsinghua University, China)
Publicity Chair:	Xiaoying Bai (Tsinghua University, China)
Financial Chair:	Leonarda Haid-Garcia (DERI, Austria)
Poster Co-chairs:	Yuting Zhao (ITC-Irst, Italy)
	Paritosh Pandya (TIFR, Italy)
Registration Chairs:	Jie Tang (Tsinghua University, China)
	Peng Wang (Tsinghua University, China)

Program Committee Members

Witold Abramowicz (Poznan University of Economics, Poland)
Dean Allemang (TopQuadrant, Inc., USA)
Chutiporn Anutariya (Shinawatra University, Thailand)
Sean Bechofer (University of Manchester, UK)
Richard Benjamins (ISOCO, Spain)
Chris Bussler (National University of Ireland, Ireland)
Enhong Chen (University of Science and Technology of China, China)
Xiaoping Chen (China University of Science and Technology, China)

Yin Chen (Hong Kong University of Science and Technology and China Southern Normal University, China)
Isabel Cruz (University of Illinois, Chicago, USA)
Mike Dean (BBN, USA)
Ying Ding (University of Innsbruck, Austria)
John Domingue (Open University, UK)
Dieter Fensel (University of Innsbruck, Austria)
Jennifer Golbeck (University of Maryland, USA)
Sung-Kuk Han (Wonkwang University, Korea)
Jeff Heflin (Lehigh University, USA)
Kaoru Hiramatasu (NTT, Japan)
Masahiro Hori (Kansai University, Japan)
Itaru Hosomi (NEC, Japan)
Jingpeng Huai (Beijing University of Aeronautics and Astronautics, China)
Mitsuru Ikeda (JAIST, Japan)
Takahiro Kawamura (Toshiba, Japan)
Yoshinobu Kitamura (Osaka University, Japan)
Ringo Lam (Wisers, Hong Kong, China)
Alain Leger (France Telecom, France)
Juanzi Li (Tsinghua University, China)
Ee-Peng Lim (Nanyang Technological University, Singapore)
Qin Lu (Hong Kong Polytechnic University, China)
Xinsheng Mao (IBM CSDL, China)
Ekawit Nantajeewarawat (Thammasat University, Thailand)
Wolfgang Nejdl (L3S and University of Hannover, Germany)
Sam-Gyun Oh (Sung Kyun Kwan University, Korea)
Jeff Pan (University of Aberdeen, UK)
Yue Pan (IBM China Research Lab, China)
Jong-Hun Park (Seoul National University, Korea)
Yuzhong Qu (SouthEast University, China)
M.R.K. Krishna Rao (KFUPM, Saudi Arabia)
Marco Ronchetti (University of Trento, Italy)
Guus Schreiber (Vrije Universiteit Amsterdam, The Netherlands)
Amit Sheth (University of Georgia and Semagix, USA)
Pavel Shvaiko (University of Trento, Italy)
Rudi Studer (University of Karlsruhe, Germany)
York Sure (University of Karlsruhe, Germany)
Hideaki Takeda (NII, Japan)
Takahira Yamaguchi (Keio University, Japan)
Yong Yu (Shanghai Jiao Tong University, China)
Michal Zaremba (National University of Ireland, Ireland)
Aoying Zhou (Fudan University, China)
Hai Zhuge (Institute of Computing Technology, Chinese Academy of Sciences, China)
Xiaoyan Zhu (Tsinghua University, China)

Additional Reviewers

Abir Qasem
Alessio Gugliotta
Alexandre Delteil
Andrew Perez-Lopez
Bangyong Liang
Barry Norton
Borys Omelayenko
Byung-Hyun Ha
Carlos Pedrinaci
Chen Wang
Christoph Tempich
Cory Henson
Daniele Turi
Dave Majernik
Dawei Hu
Denny Vrandecic
Dongmin Shin
Dong-Won Jeong
Dorene Ryder
Douglas Brewer
Fabrice Clerot
Fangkai Yang
Franck Panaget
Freddy Lecue
Gail Mitchell
Hailong Sun
Heiko Haller
Holger Lewen
Huan Li
Huiyong Xiao
Ilya Zaihrayeu
Jack Marin
Jaeyoon Jung
Jahee Kim
Jens Hartmann
Jesus Contreras
Jianxin Li
Jie Liu
Jie Tang
Jiehui Jiang
Johanna Voelker
Johanna Volker
Jose Manuel Gomez Perez
Kenta Cho
Kunal Verma
Kyung-Il Lee
Laura Hollink
Lei Zhang
Liliana Cabral
Liu Min Xing
Masumi Inaba
Matthew Perry
Max Voelkel
Maxym Mykhalchuk
Md Maruf Hasan
Mikalai Yatskevich
Min-Jeong Kim
Munehiko Sasajima
Naoki Fukuta
Nenad Stojanovic
Oscar Corcho
Peter Haase
Philipp Cimiano
Photchanan Ratanajaipan
Pinar Alper
R.K. Shyamasundar
Rachanee Ungrangsi
Roxana Belecheanu
Saartje Brockmans
Sahid Hussain
Sheng Ping Liu
Shinichi Nagano
Stefania Galizia
Steffen Lamparter
Stephan Bloedhorn
Sudhir Agarwal
Tanguy Urvoy
Tao Liu
Ted Benson
Tianyu Wo
Veronique Malaise
Vincenzo D'Andrea
Willem van Hage
Xiaoping Sun
Xin Li
Yang Yang
Yeon-Hee, Han
Yi Zhou
Yuanbo Guo
Yumiko Mizoguchi
Zhengxiang Pan
Zongxia Du

Sponsors

Golden Sponsors

European Semantic Systems Initiative

dip

Data, Information and Process Integration with Semantic Web Services

DERI

DERI INNSBRUCK

EASTWEB

Saltlux

Silver Sponsors

Knowledge Web

Sogou搜狗

Media Sponsors

W3C WORLD WIDE WEB consortium®

万维网联盟中国办事处

Table of Contents

Invited Talks

Annotation

Ontology Alignment

Document and Recommendation

Social Network and RSS

Ontology Integration and Interoperability 1

Ontology Integration and Interoperability 2

Reasoning

Application 1

Information Search

Database

Semantic Web Services 1

Semantic Web Services 2

Ontology and Tool

Application 2

Ontology and Theory

Peer-to-Peer

Industrial Track 1

Industrial Track 2

The Semantic Web: A Network of Understanding

Jim Hendler

Computer Science Department
University of Maryland
College Park, MD 20742, USA
hendler@cs.umd.edu

If you visit my Web page[1], which is not much different than most other people's in many ways, you would find many fields which are highlighted as links to other pages. In the list of my students you can find links to their pages, in the links of my papers you can find downloadable files or links to various digital libraries, and in the lists of my classes you can find links both to the Web resources I used in my classes and to University pages that describe when the classes were given, what the prerequisites were, etc. In short, a great deal of the information "on my page" is not actually on my page at all, it is provided by the linking mechanisms of the Web. It is, in fact, exactly this network effect in which I can gain advantage by linking to information created by other people, rather than recreating it myself, that makes the Web so powerful.

Now consider knowledge representation (KR). Supposing I want to create a machine-readable KR page that would contain similar information to that in my home page. I cannot get this kind of network effect using the knowledge representation techniques traditional to the AI field. First, even if I decide to use a particular representation technique, and even if it is a well-defined technique like First-order logic, there's still the issue of using information defined by someone else. My particular parser has a certain format in which it wants me to represent my information, so I write

```
ForAll(x)(Advisor(x, Hendler) -> StudentOf(Hendler,x).
```

Unfortunately, my student, John Smith, who has a knowledge base from which it could be inferred that I was his advisor, has written this as

```
Advisor(_x,_ y) :-  PhDAdvisor(_y,_x).
PhDAdvisor(Hendler,Smith).
```

When I try to unify his KB with mine, even though there is no logical mismatch, the mere syntactic differences between our representations makes it so I cannot simply use the knowledge from his KR. If the student were using a different form of KR, say some particular subset of FOL, some particular temporal logic, or some kind of modal operators, the problem would be even worse.

Even if we were using the same exact logical language, and even if we have the same implementation (so syntax matters go away), we still don't have the kind of linking we have on the Web. I cannot simply point at his KR, the way I point at someone else's web page, with the knowledge that the mechanisms of the Web will

[1] http://www.cs.umd.edu/~hendler

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 1 – 3, 2006.
© Springer-Verlag Berlin Heidelberg 2006

somehow magically get the right information for me when I click on a link (or whatever the KR equivalent might prove to be). I don't have a mechanism in most KR systems by which I could specify that a KB living somewhere else should be included into mine at query time so I could simply use the knowledge defined by someone else. In short, we don't have a way to get the network effect in KR that we get in our Web world.

In fact, in many KR systems the notion of knowledge not directly under the control of a single mechanism, incorporated at what would be the equivalent of compile time, is anathema to the design. It can lead to inconsistency in all sorts of nasty ways. For example, my student might be using knowledge in a way that is incompatible or inconsistent with mine via unexpected interactions – I said “man” implies “male” where he was using the term in the non-gendered “all men are mortals” sense. Thus, when our KBs are linked his mother becomes a male in my system, mothers are known (by me) to be female, and thus we have a contradiction leading to one of those nasty inconsistencies that causes belief revision at the least and from which all manner of nasty things could be inferred in many systems. Or consider even if we use our terms correctly, but when my query is made his server is down, and thus the list of who my students are sometimes includes him, and sometimes doesn't, again leading to potential problems.

Traditionally, the field of knowledge representation has faced these potential problems by either ignoring them (by assuming people are using the same KR system, or doing all merging at “compile” time), by addressing them as special cases (such as in the design of temporal reasoners or belief revision systems) or by defining the problem away. This latter is generally done by using inexpressive languages that don't allow inconsistency, or defining inconsistency as an “error” that will be handled offline.

Additionally, there is another issue that KR systems in AI have tended to ignore: the issue of scaling. Yes, we have often talked of algorithmic complexity, or even performance issues, but compared to the size of a good database system, or an incredible information space like the World Wide Web, KR systems have lagged far behind. The engineering challenges proposed by KBs that could be linked together to take advantage of the network effect that could be achieved thereby, are beyond the scaling issues explored in much AI work.

In short, there's a set of KR challenges that have not been widely explored until recently. First, solving syntactic interoperability problems demands standards – not just at some kind of KR logic level, but all the way down to the nitty-gritty syntactic details. Second, linking KR systems requires “extra-logical” infrastructure that can be exploited to achieve the network effect. Third, the languages designed need to be scalable, at least in some sense thereof, to much larger sizes than traditional in AI work. Fourth, and finally, achieving such linkage presents challenges to current KR formulations demanding new kinds of flexibility and addressing issues that have largely been previously ignored.

From a KR perspective, designing systems to overcome these challenges, using the Web itself for much of the extra-logical infrastructure, is the very definition of what has come to be known as the "Semantic Web." It was this thinking that led me and the other authors of a widely cited vision paper on the Semantic Web [1] to conclude that

> Knowledge representation … is currently in a state comparable to that of hypertext before the advent of the web: it is clearly a good idea, and some very nice demonstrations exist, but it has not yet changed the world. It contains the seeds of important applications, but to unleash its full power it must be linked into a single global system.

Many articles and papers have described how the Semantic Web is like traditional AI (mappings to Description logic and other formalisms, for example), but this talk will concentrate on the other side of this – the things that make Semantic Web KR different from traditional AI systems.

Reference

1. Berners-Lee, T., Hendler, J. and Lassila, O.: The Semantic Web, Scientific American 284(5) (2001) 34-43.

Transformation from OWL Description to Resource Space Model*

Hai Zhuge, Peng Shi, Yunpeng Xing, and Chao He

China Knowledge Grid Research Group,
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, 100080, China
{zhuge, pengshi, ypxing, hc}@kg.ict.ac.cn

Abstract. Semantics shows diversity in real world, document world, mental abstraction world and machine world. Transformation between semantics pursues the uniformity in the diversity. The Resource Space Model (RSM) is a semantic data model for organizing resources based on the classification semantics that human often use in understanding the real world. The design of a resource space relies on knowledge about domain and the RSM. Automatically creating resource space can relieve such reliance in RSM applications. This paper proposes an approach to automatically transform Web Ontology Language description into resource space. The normal forms of the generated resource space are investigated to ensure its normalization characteristic. The Dunhuang culture resource space is used to illustrate the approach.

1 Introduction

The machine-understandable semantics is commonly regarded as the key to the Semantic Web [3]. The W3C (www.w3.org) recommended the Web Ontology Language (OWL) in 2004 to support advanced Semantic Web applications by facilitating publication and sharing of ontology (www.w3.org/2004/OWL/).

The Resource Space Model (RSM) is a semantic model for uniformly specifying and organizing resources [16, 17]. It maps various resources (information, knowledge and services) into a multi-dimensional semantic space — a semantic coordinate system that normalizes the classification semantics. Each point in the space represents the resources of the same semantic category. Normal form and integrity theories of the RSM ensure correct semantic representation and operations [19]. The RSM theory is developed in parallel with the relational database theory [20].

The design of resource space is based on domain knowledge, application requirement and knowledge of RSM. The design method was proposed to guide the process of developing an appropriate resource space [18]. However, it still relies on designers' knowledge about domain and RSM. To relieve such reliance is an important issue of the RSM methodology.

The development of domain ontology makes codified domain knowledge. It will be very useful if we can codify the knowledge about RSM into an approach for auto-

* Keynote at ASWC2006. This work was supported by the National Basic Research Program of China (973 project No.2003CB317000) and the National Science Foundation of China.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 4 – 23, 2006.
© Springer-Verlag Berlin Heidelberg 2006

matically transforming domain ontology into resource space. The semantics in OWL description can be used to support the creation of resource space.

This paper proposes an approach to automatically transform an OWL description into a resource space to enhance the efficiency of RSM design and relieve the reliance on individual knowledge by converting individuals of OWL to resources of RSM and transforming the inheritance hierarchy relationships and properties of resources into axes of RSM.

Relevant work includes the transformation between OWL service and the Unified Modeling Language (UML) [7], the converting from OWL ontology to UML [6], the bidirectional mapping between Attempto Controlled English (ACE) and OWL [10], converting from OWL DLP statements to logic programs [12], and the method for converting the existing thesauri and related resources from native format to RDF(S) or OWL [1]. A method reflecting the taxonomic relationship of products and service categorization standards (PSCS) in an OWL Lite ontology was proposed [9].

Related work also concerns software engineering area. The structural software development can be regarded as a multiple step transformation from the semantic specification on domain business into the semantic specification on software. Semantic specification tools like the Entity-Relationship model help developers transform domain business into relational model [4, 13]. The transformation from the E-R model into the relational database was investigated [2, 5, 15].

2 The Synergy of the Semantics in Real World, Document World and Abstraction World

Real world semantics used by human is hard to be understood by machines. Modeling languages like UML are for specifying real world semantics in standardized symbol systems.

Semantics in the mental world can be intuitive or abstract. Abstract semantics takes the form of symbolized and geometrical principles and theories. Human often use classification method to recognize the real-world. The implementation of the classification-based RSM depends on the data structures in the machine world, while the display of a resource space can be in the geometrical form of the abstraction world.

Semantics in the machine world is hard for ordinary people to understand. The XML, RDF and OWL mediate the machine world and the document world at different semantic levels.

Different semantics overlap and interact with each other to establish and develop the interconnection semantics as shown in Fig. 1. The future interconnection environment needs the synergy of the diversity and uniformity of semantics in the real world, the document world and the mental abstraction world. Automatic transformation between semantics of different levels is an important issue. The transformation from an OWL description to the RSM generalizes the semantics in the machine world and the document world. Since RSM is based on classification semantics, the created resource space (called OWL-based resource space) does not keep all the semantics described in OWL file. Transformations from OWL into abstract SLN and from UML into OWL are also significant.

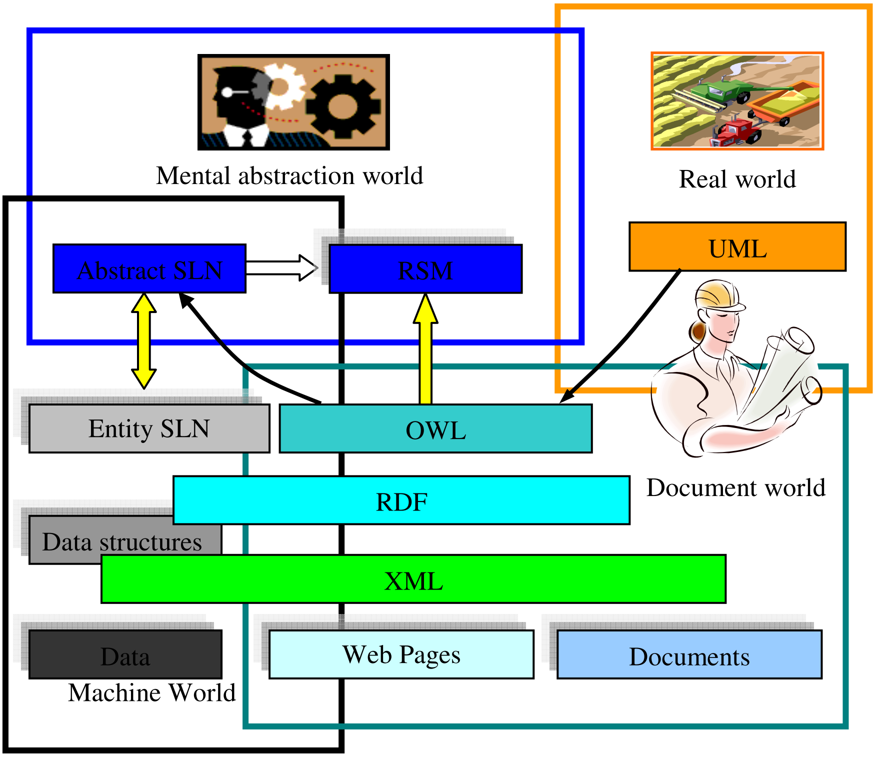


Fig. 1. The synergy of the semantics of four worlds in future interconnection environment

3 Basic Elements of OWL and an Example of RSM

Ontology facilitates the uniformity and sharing of domain knowledge by five types of basic modeling primitives: classes or concepts, relations, functions, axioms and instances [8, 14]. OWL provides three increasingly expressive sublanguages designed for specific users. OWL Lite is for the users who primarily need classification hierarchy and simple constraint features. OWL DL supports the maximum expressiveness without losing computational completeness and decidability of reasoning systems. OWL Full supports maximum expressiveness and the syntactic freedom of RDF without computational guarantees.

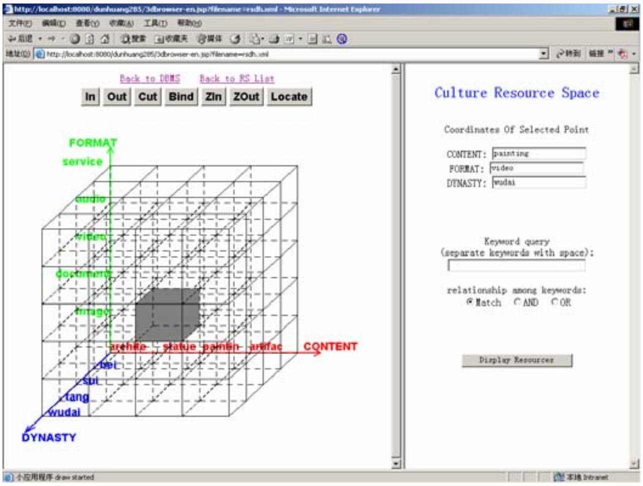
The following are basic elements of OWL:

(1) *Class* defines a class. An individual is an instance of a class.

(2) *rdfs*:*subClassOf* specifies the subclass relation.

(3) Properties are owned by classes or instances and divided into two types: *ObjectProperty* and *DatatypeProperty*. *ObjectProperty* specifies the relation between two instances, which belong to the same or different classes. *DatatypeProperty* indicates the relation between instance and RDF literals or XML Schema datatypes such as string, float or date.

(4) *rdfs*:*subPropertyOf* represents the inheritance of properties.



Culture Resource Space

Coordinates Of Selected Point

CONTENT: painting
FORMAT: video
DYNASTY: wudai

Keyword query
(separate keywords with space):

relationship among keywords:
● Match ○ AND ○ OR

Fig. 2. The three-dimensional resource space browser

(5) *rdfs*:*domain* and *rdfs*:*range* restrict the anterior and posterior values of a property respectively. There are also some characteristics and restrictions, such as *TransitiveProperty*, *SymmetricProperty*, *allValuesFrom* and *Cardinality*, for describing property.

The following elements in OWL are used to improve the ability of describing the relations between classes, individuals and properties.

(1) *equivalentClass* and *equivalentProperty* represent the equivalence between classes and properties respectively.
(2) *sameAs* indicates that two individuals with different names are logically equal.
(3) *differentFrom* and *AllDifferent* explicitly distinguish one individual from others.
(4) *intersectionOf*, *unionOf* and *complementOf* are for set operation. They usually represent how a class is composed by other classes.
(5) *disjointWith* prevents a member of one class from being that of another class.

These elements help the mapping between different ontologies and describe more complex relationships between classes and individuals.

The most commonly used resource space is two- or three-dimensional, which can be displayed on screen and manipulated by users with ease. Fig.2 shows a three-dimensional resource space browser for Dunhuang culture exhibition. Resources can be located and manipulated by moving the black cube representing a point in the space. The black cube can be controlled by moving mouse and clicking the “In” and “Out” buttons.

4 Transformation from OWL Description into RSM

4.1 Process of Transformation

The main process of creating resource space from OWL file is shown in Fig.3. The input consists of the ancestor classes and the OWL file. The ancestor classes are the top-level classification of resources in an application.

The first step is to eliminate synonym in OWL file as the *equivalentClass*, *equivalentProperty* and *sameAs* in OWL may cause classification confusion when creating a resource space. A solution is to use one complex name to replace the synonyms.

Some individuals in OWL are transformed into resources in resource space. The inheritance hierarchy relationships and properties of resources in OWL are converted into axes of the OWL-based resource space.

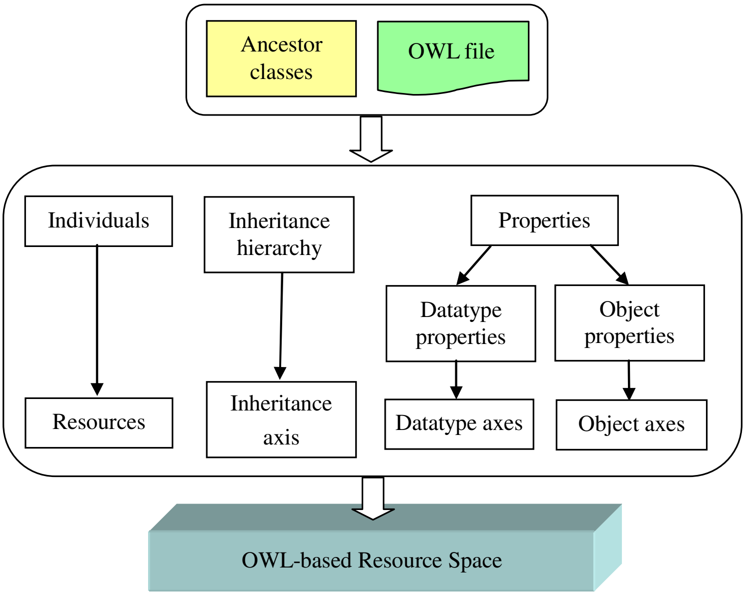


Fig. 3. The main process of creating resource space from OWL file

4.2 From Individuals to Resources

The individuals belonging to the ancestor classes in OWL file can be transformed into resources in resource space. The following are examples of two individuals:

```
<BMP rdf: ID="instance_0001">
   <NAME            rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
   cave305.bmp
   </NAME>
   <AUTHOR rdf:resource="Mr.Zhao"/>
   ……
</BMP>
<RESEARCHER rdf: ID="Mr.Zhao">
    <AGE        rdf:datatype="http://www.w3.org/2001/XMLSchema#unsignedInt">
    40</AGE>
    ……
</RESEARCHER>
```

The individual "instance_0001" is an instance of class “BMP”. It owns two properties “NAME” and “AUTHOR”. “Mr.Zhao” is also an individual which is instantiated from the class “RESEARCHER”. Here the input ancestor class is “File”. Only individuals inherited from it are regarded as resources. Therefore, the former individual is a resource. The latter indicates the value of property “AUTHOR” of “instance_0001”, so it will not be regarded as a resource in this example.

4.3 From Inheritance Hierarchy to Inheritance Axis

The ancestor classes inherently represent classification. This corresponds to the classification principle of RSM, so inheritance hierarchy of resources in OWL will be transformed into an axis named inheritance.

The process of forming inheritance axis consists of three steps:

(1) Parse the OWL file to find the subclasses and instances of every input ancestor class; and,

(2) Form a hierarchical structure like Fig. 4 according to their inheritance relationships: node represents class or instance and edge represents the inheritance relationship between classes or the instance relationship between instance and its class. So each edge starts from a class and ends at its super class or starts from an instance and ends at its class. The hierarchical structure is a top-down directed graph with ancestor classes at the top level and the instances (resources) at bottom level.

(3) Transform the structure into a tree or trees, which are taken as the coordinates.

In Dunhuang culture resource space, the class “File” is the input ancestor class to create the file resource space. It has four subclasses: “document”, “image”, “audio” and “video”, which also have their own subclasses. The subclasses of “document” are “PDF” and “TXT”. They indicate different types of files. Their declarations in Dunhuang OWL file are as follows:

```
<owl:Class rdf:ID="File"/>
<owl:Class rdf:ID="image">
        <rdfs:subClassOf rdf:resource="File"/>
  </owl:Class>
 <owl:Class rdf:ID="JPEG">
         <rdfs:subClassOf rdf:resource="image"/>
  </owl:Class>
  <owl:Class rdf:ID="BMP">
         <rdfs:subClassOf rdf:resource="image"/>
  </owl:Class>
  <owl:Class rdf:ID="document">
        <rdfs:subClassOf rdf:resource="File"/>
  </owl:Class>
  <owl:Class rdf:ID="PDF">
        <rdfs:subClassOf rdf:resource="document"/>
  </owl:Class>
  <owl:Class rdf:ID="TXT">
        <rdfs:subClassOf rdf:resource="document"/>
```

```
</owl:Class>
<owl:Class rdf:ID="video">
    <rdfs:subClassOf rdf:resource="File"/>
</owl:Class>
<owl:Class rdf:ID="AVI">
    <rdfs:subClassOf rdf:resource="video"/>
</owl:Class>
<owl:Class rdf:ID="SWF">
    <rdfs:subClassOf rdf:resource="video"/>
</owl:Class>
<owl:Class rdf:ID="audio">
    <rdfs: subClassOf rdf: resource="File"/>
</owl:Class>
<owl:Class rdf:ID="MP3">
    <rdfs:subClassOf rdf:resource="audio"/>
</owl:Class>
<BMP rdf:ID="instance_0001">… </BMP>
<TXT rdf:ID="instance_0011">…</TXT>
<AVI rdf:ID="instance_0021">… </AVI>
```

The inheritance structure in Fig.4 includes the ancestor class, its offspring classes and these instances (resources). The top level is “File”. The second level includes “document”, “image”, “video” and “audio”. The lower level includes “PDF”, “TXT”, “JPEG”, “BMP”, “MP3”, “AVI” and “SWF”. The instances are resources at the bottom level. The resource “instance_0001”, “instance_0011” and “instance_0021” are instances of “BMP”, “TXT” and “AVI” respectively.

If the inheritance hierarchy is a tree or trees, it can be transformed into an axis to represent the category of resources. This axis is called inheritance axis. The elements in the hierarchy are the coordinates on inheritance axis except for the instances. If the hierarchy is a tree, there is only one node at the top level. This node will not be a coordinate on inheritance axis because it cannot classify resources. A resource’s coordinates on this axis is composed of its class and ancestor classes together. The coordinates at different levels represent different scales of classification. In Dunhuang resource space, the inheritance hierarchy is a tree, so it can be transformed into an inheritance axis directly named “Format”. As shown in Fig.5, the top level coordinates on this axis include “document”, “image”, “audio” and “video”. The second level coordinates are “TXT”, “PDF”, “JPEG”, “BMP”, “MP3”, “AVI” and “SWF”. The coordinate of “instance_0011” on this axis is “document.TXT”, where the dot separates coordinates on different scales.

OWL supports multiple-inheritance, that is, the inheritance structure is not a tree but a graph. The graph should be converted into tree(s) because the coordinates on an axis in RSM should be tree(s). Algorithm 1 converts an inheritance graph into tree(s). Multiple-inheritance indicates that one subclass inherits from two or more classes. It implies that the parent classes cannot classify the resources independently. So the algorithm eliminates the nodes of parent classes and linked edges directly from the hierarchy. The subclasses are reserved as the top level elements of the derived tree.

This algorithm guarantees that the output tree contains the resources and their parent classes at least. Here the function outdegree() and indegree() are used to get the outdegree and indegree of node in a graph respectively. setMark() and getMark() are for setting or getting markers of nodes. getParent() is for getting the parent class of a node in a graph. getUntreatedNumber() is for getting the number of untreated nodes in T.

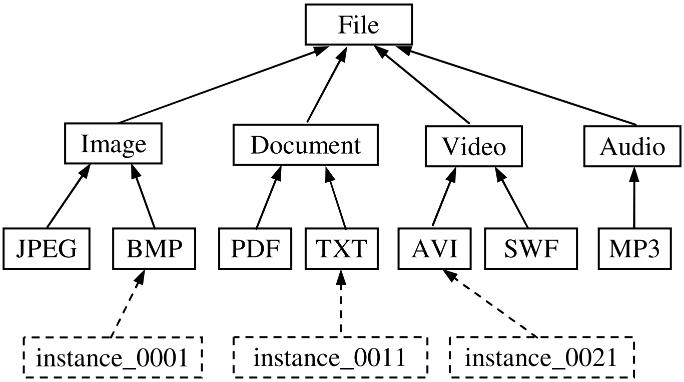


Fig. 4. The inheritance hierarchy of resources in Dunhuang application. The solid and dashed rectangles indicate classes and instances respectively. The solid and dashed arrows indicate inheritance relationships and instance relationships respectively.

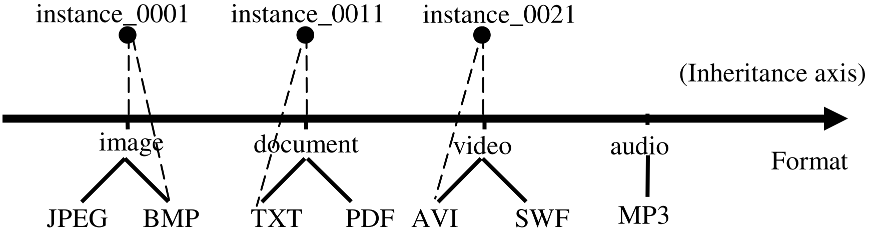
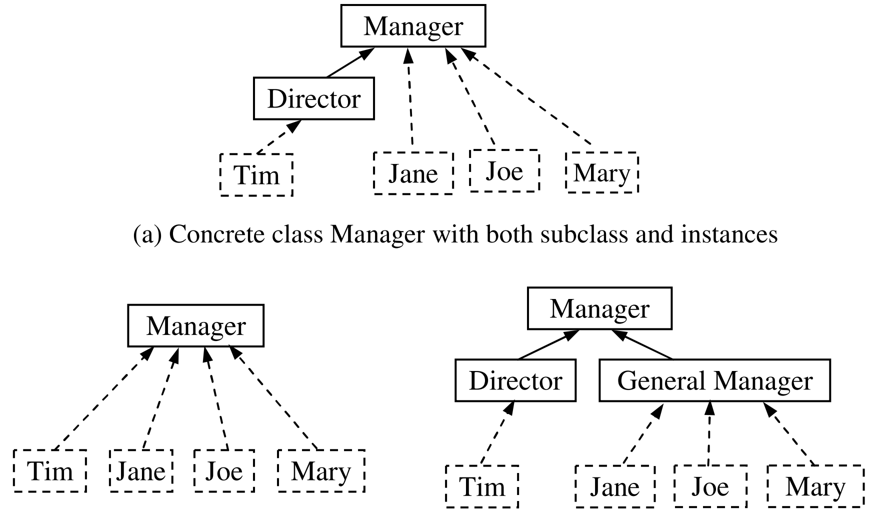


Fig. 5. The inheritance axis of Dunhuang resource space

```
Algorithm 1. void GraphToTree(Graph G , Tree T)
{/*convert a connected directed graph G into a tree(s)
T*/
  For every node
  {/*treat from the bottom level*/
    If( outdegree(node, G) = 0 ) {/*a bottom node*/
      Output node into T as a leaf;
      If( indegree(node, G) = 0 ) {
        Show message "error: an individual hasn't
class";
        Return;
      }
      Else if( indegree(node, G)=1 ) {/*uni-
inheritance*/
        setMark(node, T, treated);
        Output getParent(node,G) into T;
      }
    }
  }
  While( getUntreatedNumber(T)>0 ) {
    Get an untreated node from T;
    If( indegree(node, G) = 1 ) {/*qualified node*/
      If( getMark( getParent(node,G) ) != deleted ) {
        Output getParent(node, G) as parent of node
into T;
      }
    }
    Else if( indegree(node, G)>1 ) {/*multi-
inheritance*/
      For every ancestor of node in G {
        If( getMark(ancestor, G) != deleted ) {
          setMark( ancestor, G, deleted );
          Delete ancestor from T;
        }
      }
    }
    setMark( node, T, treated ); /*mark treated node*/
  }
}
```

In OWL, concrete classes may own subclasses and instances, but abstract classes can only have subclasses. For a concrete class that has both subclasses and instances, it is possible that the instances of the concrete class cannot be located by the subclass coordinates. For example in Fig.6 (a), the concrete class "Manager" has a subclass "Director" and three instances "Jane", "Joe" and "Mary". "Director" has its own instance "Tim". If this structure is converted into coordinates of the inheritance axis, the coordinates include "Manager" and "Director" at two levels. The coordinate "Director" can only specify "Tim", but cannot specify "Jane", "Joe" or "Mary". There are two strategies to deal with this kind of concrete classes: (1) discard its subclasses and combine the instances of subclasses into it; (2) add a new subclass for the concrete class, instances of the concrete class can be identified as instances of the subclass added. The former strategy weakens the classification semantics of resources but simplifies the process. The latter enhances the classification semantics.

Algorithm 2 is to check and deal with concrete classes. The parameter bDiscard distinguishes the two strategies. If bDiscard = true, subclasses are discarded. Otherwise a new subclass is added whose name is provided by the parameter newClassName. Fig.6 (b) is the result when bDiscard=true. The subclass "Director" is deleted and its instance "Tim" becomes the instance of "Manager". Fig.6 (c) shows the result when bDiscard=false. A new subclass named "General Director" is added with "Jane", "Joe" and "Mary" as its instances. The result is transformed into the coordinates on the inheritance axis.



(a) Concrete class Manager with both subclass and instances

(b) Result of discarding subclasses

(c) Result of adding subclass

Fig. 6. An example of processing concrete class

```
Algorithm 2. Boolean CheckAndChangeConcreteClass (Class
conClass, Boolean bDiscard, String newClassName ) {
  If(conClass has both instances and subclasses) {
    If(bDiscard ) { /*discard subclasses*/
      For every subclass of conClass {
        Move its instances into conClass;
        Delete subclass;
      }
    }
    Else{ /*add a new subclass*/
      Create a new class named newClassName;
      Get all instances of conClass;
      Move the instances into newClassName;
      Add newClassName as a subclass of conClass;
    }
    Return true;
  }
  Else{/*need not be modified*/
    Return false;
  }
}
```

Then an inheritance axis is created according to inheritance relationships of resources and their ancestor classes. There is only one inheritance axis in the OWL-based resource space. The hierarchical coordinates provide users with multiple scale location according to application requirements.

4.4 From Properties to Property Axes

Properties in OWL are used to describe characteristics of classes and individuals. They can be adopted as classification principles of resources in RSM as well. If the domain of a property includes the ancestor classes, it can be transformed into a property axis. The property is called source property of the axis.

"DatatypeProperty" declares a property with one of data types, which come from RDF literals and XML Schema data types. The property value belongs to the specified datatype. A datatype property can be converted into an axis called datatype axis in OWL-based resource space. The axis is named after the property name and its coordinates include all elements within the property value range. If the range of datatype property is only specified as a kind of datatype without other restrictions, the elements in the range may be finite (such as "boolean" type) or infinite (such as "string" and "int" type). Because the coordinates on an axis must be finite, the unqualified property should classify their values into finite classes at first. Different datatypes use different strategies to classify its infinite elements into finite classes so as to ensure no intersection between classes. For example, "string" may be classified according to alphabet order. The classification strategy may contain hierarchical structure to classify resources with different scales. But the fixed classification approach classifies various resources into the same classes. Other classification methods in pattern recognition and text processing can be adopted to classify resources according to their characteristics. The restrictions of datatype property range are allowed in OWL. For convenience, Dunhuang OWL imports "xsp.owl" developed by Protégé to restrict data types. For example, Dunhuang resources have an "unsignedInt" type property "CaveNo" to specify which cave the resources reside in. Its declaration is as follows:

```
<owl:DatatypeProperty rdf:ID="CaveNo">
    <rdfs:domain>
        <owl:Class rdf:resource="File"/>
    </rdfs:domain>
    <rdfs:range>
        <rdfs:Datatype>
            <xsp:base
            rdf:resource="http://www.w3.org/2001/XMLSchema#unsignedInt"/>
            <xsp:minInclusive
            rdf:datatype="http://www.w3.org/2001/XMLSchema#unsignedInt">
            1</xsp:minInclusive>
            <xsp:maxInclusive
            rdf:datatype="http://www.w3.org/2001/XMLSchema#unsignedInt">
            900</xsp:maxInclusive>
        </rdfs:Datatype>
    </rdfs:range>
</owl:DatatypeProperty>
```

The domain of "CaveNo" is the class "File". The range of "CaveNo" is restricted from 1 to 900. Then the property can be transformed into an axis. Its coordinates are the elements within the property range. The resources and their property values are described as follows.

```
<BMP rdf: ID="instance_0001">
  <CaveNo>305</CaveNo>
  ……
 </BMP>
 <TXT rdf:ID="instance_0011">
  <CaveNo>220</CaveNo>
  ……
</TXT>
<AVI rdf: ID="instance_0021">
  <CaveNo>530</CaveNo>
……
</AVI>
```

The values of “instance_0001”, “instance_0011” and “instance_0021” are 305, 220 and 530 respectively.

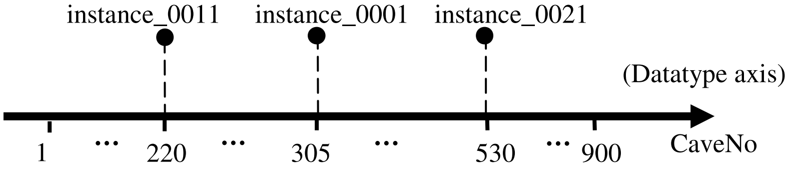


Fig. 7. The axis transformed by datatype property CaveNo

Fig. 7 is the axis named after the property “CaveNo”. Its coordinates include all the cave numbers in Dunhuang from 1 to 900. A resource’s coordinate on this axis is its property value. The coordinate of “instance_0001” on this axis is 305. So a datatype property can be transformed into a data type axis.

In OWL, an object property is a relation between two objects and declared by “ObjectProperty”. An object property can be transformed into a homonymous axis, called object axis. Its coordinates consist of the ancestor classes of elements within the property’s range and they are usually in inheritance hierarchy. A resource’s coordinate on object axis is composed of the ancestor classes of its property’s value. All the elements within the property’s range, with their ancestor classes, form an inheritance hierarchy. The procedure of creating object axis is similar to that of creating inheritance axis. Algorithm 1 and algorithm 2 are also used to get a directed tree or trees. The output tree structure is converted into coordinates on object axis. For example, an object property “Content” is defined in Dunhuang OWL file as follows:

```
<owl:ObjectProperty rdf:ID="Content">
  <rdfs:domain rdf:resource="File"/>
  <rdfs:range rdf:resource="ContentClass"/>
</owl:ObjectProperty>
```

“Content” describes the content represented by Dunhuang resources. Its domain is “File” class and its range is “ContentClass”. The values of resources on this property are described as follows:

```
<BMP rdf:ID="instance_0001">
    <Content rdf:resource="cave_305"/>
    …
 </BMP>
<TXT rdf:ID="instance_0011">
      <Content rdf:resource="story_1"/>
      …
</TXT>
<AVI rdf:ID="instance_0021">
        <Content rdf:resource="statue_530_2"/>
        …
</AVI>
```

The range of this property is declared as class “ContentClass”. Its subclasses are “painting”, “statue” and “architecture”. They also have their own subclasses, such as “flyer”, “story”, “separate”, “attached” and “cave”. “story_1”, “statue_530_2” and “cave_305” are the instances of “story”, “separate” and “cave” respectively. The values of resource “instance_0001”, “instance_0011” and “instance_0021” of property "Content" are “cave_305”, “story_1” and “statue_530_2” respectively. The inheritance hierarchy of "ContentClass" is given in Fig. 8.

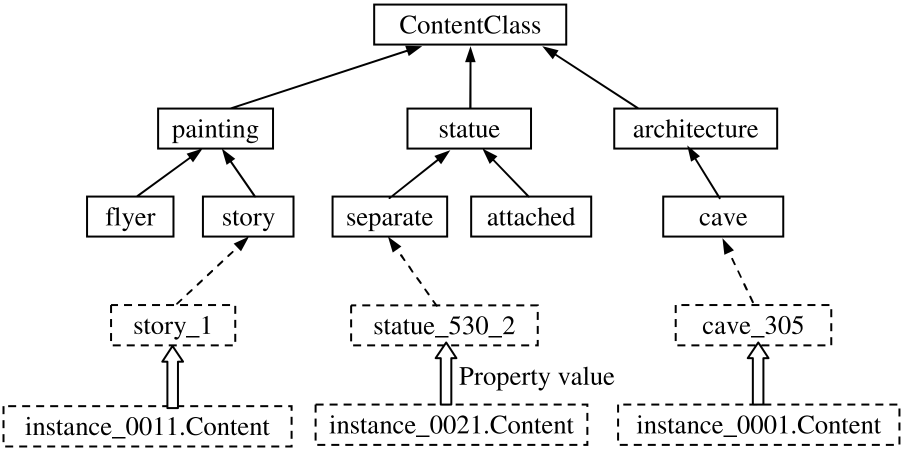


Fig. 8. The inheritance hierarchy of class “ContentClass” and the values of resources. Here the solid and dashed rectangles indicate classes and instances respectively. The solid arrows indicate the inheritance relations.

Based on the structure, an object axis named “Content” is created and shown in Fig.9. “ContentClass” is not transformed into a coordinate because there is only one element at the top level. The coordinates of “instance_0001”, “instance_0011” and “instance_0021” are “architecture.cave”, “painting.story” and “statue.separate” respectively.

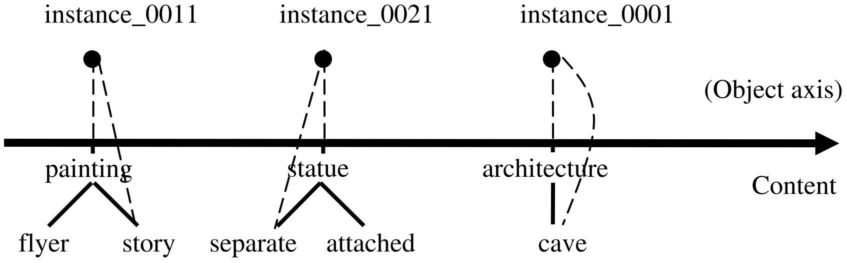


Fig. 9. Object axis derived from object property Content

In fact, not any property of ancestor classes in OWL should be transformed into a property axis in OWL-based resource space. For example, every Dunhuang resource has a property “NAME”. Suppose that “NAME” is converted into an axis “NAME”. Then each coordinate on it can only identify one resource if name duplication is prohibited. Axes generated from this kind of properties hardly classify resources efficiently. It is not an easy job to judge if a property should be transformed into an axis because it depends on the classification semantics represented by the property. It may need user’s analysis and choice. The ratio of resource number to coordinate number on the axis can be used as a referenced principle: if the ratio is close to 1, the property should not be transformed.

4.5 Combination of Axes and Resources into Resource Space

Resources and axes derived from OWL are combined to form a coordinate system. Every resource has a location determined by their ancestor classes and property’s values in this coordinate system. They are inserted into corresponding points in the space. A point in the space uniquely represents a set of resources. From the definition of resource space [16], this coordinate system constitutes a resource space. The structure of the Dunhuang OWL-based resource space is shown in Fig. 10. For simplification, only the coordinates on one layer in the coordinate hierarchy are shown.

There are three axes named “Format”, “Content” and “CaveNo” respectively. The “Format” axis is the inheritance axis derived from the inheritance hierarchy of resources. The “CaveNo” is a datatype axis and directly comes from the homonymous datatype property. The object axis “Content” is transformed from the same name object property. Every resource in the space is specified by a tuple of coordinates. For instance, the resource "instance_0001" corresponds to the point (architecture, image, 305). That means the resource is an image file and describes the architecture of the 305th cave.

The created resource space includes an inheritance axis and several property axes, which are transformed from the inheritance hierarchy and properties of resources respectively. The resources are derived from individuals and inserted into the resource space according to their ancestor classes and property values.

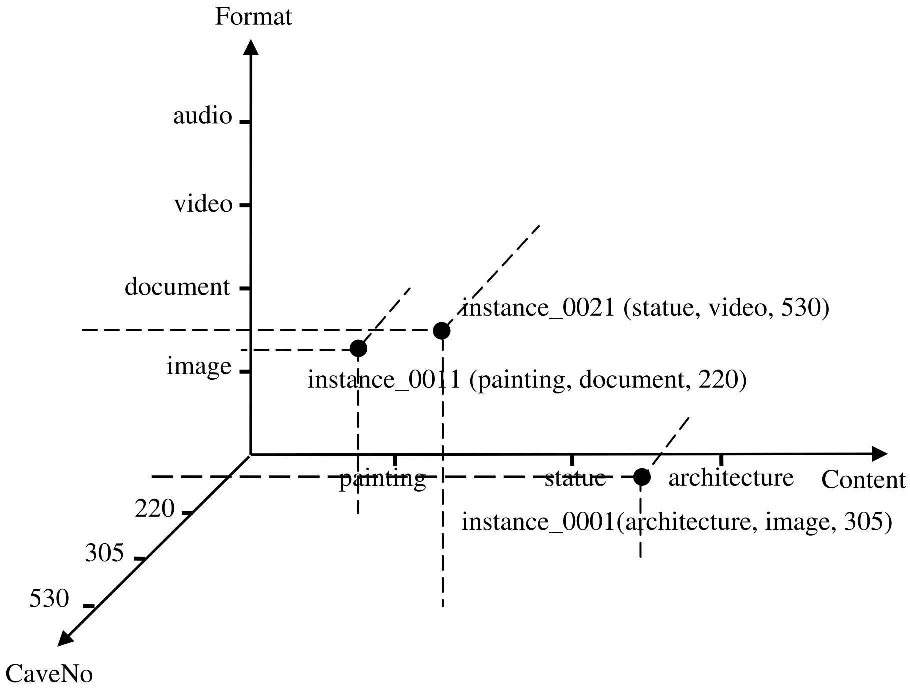

Format
audio
video
document
image
instance_0021 (statue, video, 530)
instance_0011 (painting, document, 220)
painting
statue
architecture
Content
instance_0001(architecture, image, 305)
220
305
530
CaveNo

Fig. 10. The simplified resource space derived from the Dunhuang OWL file

5 Analysis of Normal Forms

Normal forms guarantee the correctness of operations on RSM [17, 20]. Normal forms of the resource space generated from OWL file are important for their successful application. Here we assume that the OWL file is well-defined (i.e., the file can represent domain knowledge correctly and clearly) so that the resource space can represent correct classification semantics. The coordinates on an axis can be hierarchical, but usually the coordinates at the same level can satisfy certain demand of application. The hierarchical coordinates can be mapped into flat coordinates by only projecting the same level coordinates onto the axis. So here only considers the flat case for simplification.

5.1 The First Normal Form

The first-normal-form (1NF) of resource space requires that there is no name duplication between coordinates at any axis. The 1NF can be easily checked by comparing all coordinates on one axis. The unqualified resource spaces can be upgraded to 1NF after combining the duplicate coordinates into one and the corresponding resources into one set.

A well-defined OWL file does not contain duplicated classes, instances and property values. So the coordinates consisting of classes, instances and property values at any axis should not be duplicated. Hence the OWL-based resource space satisfies 1NF.

5.2 The Second Normal Form

The second-normal-form (2NF) of a resource space is a 1NF, and for any axis, any two coordinates are independent from each other. The 2NF avoids implicit coordinate duplication, and prevents one coordinate from semantically depending on another. In the application, the implicit duplication and semantic dependence are concerned with the domain knowledge. Here the semantic independence means that a coordinate is not the synonym, abstract concept, specific concept, instance or quasi-synonym of another coordinate.

Since the synonymic classes, properties and individuals are already combined during preprocessing, there are no synonymic coordinates on the inheritance axis and property axes. In a well-defined OWL file, the abstract concept of a coordinate should be declared as its ancestor class. Because the hierarchical structure of coordinates is based on inheritance relations, the abstract concept and the coordinates are at different levels. So there is no abstract concept of a coordinate at the same level. The specific concept and instance of a coordinate should be its subclass and instance respectively. They are also at different levels in the hierarchical structure of coordinates. In order to avoid semantic confusions, the coordinates at different levels should not be used at the same time. During the procedure of creating inheritance axis, the multi-inheritance problem is solved. So every resource has a certain value on axis. The quasi-synonymic classes do not influence the resource classification. On the datatype axis, the coordinates are one type of values or their classification. The quasi-synonymic values cannot influence the resource classification because a resource's coordinate is a certain value. On an object axis, the resource coordinates are ancestor classes of their property values. The coordinates on object axis are similar to those on the inheritance axis. They can avoid classification confusion of resources. So there are no influential quasi-synonyms on any axis.

The 2NF avoids the intersection of resource sets on different coordinates. In the resource space created from a well-defined OWL file, the resources are classified clearly by the coordinates on any axis. So the coordinates on any axis are semantically independent. Generally, the classification confusion on axis implies that the OWL file contains some confusing description and it should be modified to prevent semantic confusion. In other words, a well-defined OWL file can be directly transformed into a resource space satisfying the 2NF.

5.3 The Third Normal Form

A 2NF resource space is 3NF if any two axes are orthogonal with each other [16]. From the generation process, we know that an OWL-based resource space contains one inheritance axis and several property axes. Then, we have the following lemmas:

Lemma 1. In the OWL-based resource space, any two axes are orthogonal, if and only if: (1) the inheritance axis is orthogonal to any property axes, and (2) any two property axes are orthogonal.

Lemma 2. The orthogonality between two axes is transitive, that is, if X$\perp$X' and X $\perp$X'', then X'$\perp$X'' [17].

Lemma 3. In the OWL-based resource space, if the inheritance axis is orthogonal to any property axes, any two property axes are orthogonal with each other.

Proof. Let the inheritance axis be X^I, and, X_1^P and X_2^P be two arbitrary property axes. If the inheritance axis is orthogonal with any property axis, $X^I \perp X_1^P$ and $X^I \perp X_2^P$ hold. Because $X^I \perp X_1^P \Leftrightarrow X_1^P \perp X^I$ and according to Lemma 2, $X_1^P \perp X_2^P$ holds, i.e., two property axes are orthogonal. □

Theorem 1. If the OWL-based resource space is in 2NF and the inheritance axis is orthogonal with any property axes, the resource space satisfies 3NF.

Proof. From Lemma 3, if the inheritance axis is orthogonal with any property axes, we have: any two property axes are orthogonal.

From Lemma 1, we have: any two axes are orthogonal in the OWL-based resource space.

According to the definition of 3NF, the resource space is in 3NF. □

Lemma 4. For two axes X_i and X_j in a resource space, $X_j \perp X_i \Leftrightarrow R(X_j) = R(X_i)$ holds [20].

Theorem 2. In a 2NF OWL-based resource space, its inheritance axis is denoted as X^I. If $R(X^I) = R(X^P)$ holds for any property axis X^P, the resource space satisfies 3NF.

Proof. From Lemma 4, $R(X^I) = R(X^P)\quad X^I \perp X^P$.

Then the inheritance axis is orthogonal with any property axis.

According to Theorem 1, the resource space satisfies 3NF. □

Lemma 5. If a resource r owns the property P, then r can be represented by the property axis X^P transformed from P, that is, $r \in \mathrm{R}(X^P)$.

Proof. According to the generation process of property axis, the coordinates on X^P may consist of three kinds of elements: all elements within the range, a classification of all elements in the range or the ancestor classes of all the elements within the range. Because r owns the property P, so the P's value of r is within the range, r has a coordinate on X^P. So $r \in R(X^P)$ holds. □

Theorem 3. If every property axis of the 2NF resource space RS is transformed from the common property (the property owned by all the ancestor classes of resources), the resource space RS satisfies 3NF.

Proof. Let E_R be the universal resources to be organized by RS, X^I be the inheritance axis and X^P be an arbitrary property axis. We can get $R(X^I) \subseteq E_R$ and $R(X^P) \subseteq E_R$.

For any resource r, we have $r \in E_R$.

(1) Since r is an instance of a class, r can find its ancestor class on X^I.

Then $r \in R(X^I)$ and $E_R \subseteq R(X^I)$ hold.

From $R(X^I) \subseteq E_R$, we can get $R(X^I) = E_R$.

(2) Since r is an instance of a class, it has the same properties of its ancestor class. P is a common property and owned by every ancestor class.

Then, we have: r must own P as its property.

From Lemma 5, $r \in R(X^P)$ holds.

Because $r \in E_R$ holds, we can get $E_R \subseteq R(X^P)$.

And from $R(X^P) \subseteq E_R$, then we have: $R(X^P) = E_R$ holds.
From (1) and (2), we get $R(X^I) = R(X^P)$.
Then according to Theorem 2, *RS* satisfies 3NF.

From Theorem 3, we know that if every axis in OWL-based resource space is created by a common property, then the resource space satisfies 3NF. Therefore the algorithm using this condition can generate a 3NF resource space.

6 Strategy and Discussion

Integration of OWL files developed by team members is very important in ontology engineering. Assume that OWL-file is the integration of OWL-file1 and OWL-file2 denoted as OWL-file=OWL-file1$\cup$ OWL-file2, and that RS, RS1 and RS2 are resource spaces created from OWL-file, OWL-file1 and OWL-file2 respectively. If the integration operation $\cup$ is defined according to the union of graphs, then it does not reduce resources, properties and classes, therefore RS_1 and RS_2 are the subspaces of RS (i.e., all resources, axes and coordinates in RS1 or in RS2 are also in RS). If there exist common axes between RS_1 and RS_2, then RS_1 and RS_2 can be integrated by join operation: $RS_1 \cdot RS_2$ [16, 17]. Since join operation does not increase any new axis, coordinate and resource, $RS_1 \cdot RS_2$ is also a subspace of RS. This tells us a strategy of transformation from OWL into resource space: *Integrate OWL files rather than resource spaces*, that is, select the integrated OWL file for transformation to reserve more semantics rather than select the individual OWL files for transformation and then integrate the created resource spaces.

The RSM can accurately locate resources and has a firm theoretical basis for ensuring the correctness of resource operations. A two-dimensional or three-dimensional resource space can be easily displayed, manipulated and understood in mental abstraction world. Higher-dimensional resource space needs to be split into several lower dimensional resource spaces by the split operation for display [16]. But its implementation depends on the underlying structure in the machine world.

The OWL is being widely accepted by researchers and ontology developers. There will be rich OWL-based ontologies, which are the basis of automatically generating the RSM. The OWL is not designed for human to read so it is hard for human to maintain it. The OWL needs to develop its theoretical basis.

Integrating OWL with RSM can obtain advantages and overcome shortcomings of both. One strategy is to place the RSM at the high level for efficient locating and effective management of resources and place the OWL description at the low level to provide ontology support. The underlying ontology supports the normalization of the RSM [16]. The join and merge operations of RSM support the management of multiple resource spaces which could be generated from the same OWL file.

7 Conclusion

This paper investigates the semantics of the interconnection environment, and proposes an approach to automatically create a resource space from a given OWL file, and analyzes the normal forms of the OWL-based resource space. This approach can make use of existing ontology and relieve the dependence on developers' knowledge. The integration of RSM and OWL can obtain advantages of both. Strategies for transformation and integration are given. Ongoing work is the transformation between OWL and other forms of semantics like the semantic link network SLN [17].

Acknowledgement

The authors thank all team members of China Knowledge Grid Research Group (http://www.knowledgegrid.net) for their help and cooperation.

References

1. Assem, M., Menken, M. R., Schreiber, G., Wielemaker, J., Wielinga, B. J.: A Method for Converting Thesauri to RDF/OWL, International Semantic Web Conference, Hiroshima, Japan, (2004) 17-31.
2. Batini, C., Ceri, S., Navathe, S. B.: Conceptual Database Design: an Entity-Relationship Approach, Benjamin and Cummings Publ. Co., Menlo Park, California, 1992.
3. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American*, 284(5) (2001) 34-43
4. Chen, P. P.: The Entity-Relationship Model, Towards a Unified View of Data, *ACM-Transactions on Database Systems*, 1 (1) (1976) 9-36.
5. Embley, D. W.: Object Database Development Concepts and Principles, *Addison Wesley*, 1997.
6. Gaševic, D., Djuric, D., Devedžic, V., Damjanovic, V.: Converting UML to OWL Ontologies, *Proceedings of the 13th International World Wide Web Conference*, NY, USA, (2004) 488-489.
7. Grønmo, R., Jaeger, M. C., Hoff, H.: Transformations between UML and OWL-S, *the European Conference on Model Driven Architecture Foundations and Applications (ECMDA-FA)*, Springer-Verlag, Nuremberg, Germany, November, 2005.
8. Gruber, T. R.: A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition*, 5 (2) (1993) 199-220.
9. Hepp, M.: A Methodology for Deriving OWL Ontologies from Products and Services Categorization Standards, *Proceedings of the 13th European Conference on Information Systems (ECIS2005)*, Regensburg, Germany, (2005), 1-12.
10. Kaljurand, K.: From ACE to OWL and from OWL to ACE, *The third REWERSE annual meeting*, Munich, March, 2006.
11. Marca, D., McGowan, C.: SADT: Structured Analysis and Design Techniques, McGraw-Hill, 1987.
12. Motik, B., Vrandecic, D., Hitzler, P., Sure, Y., Studer, R.: dlpconvert - Converting OWL DLP Statements to Logic Programs, *System Demo at the 2nd European Semantic Web Conference,* Iraklion, Greece, May, 2005.
13. Ng, P. A.: Further Analysis of the Entity-Relationship Approach to Database Design, *IEEE Transaction on Software Engineer*, 7(1) (1981) 85-99.
14. Neches, R., Fikes, R. E., Gruber, T. R., Patil, R., Senator, T., Swartout, W.: Enabling Technology for Knowledge Sharing, *AI Magazine*, 12 (3) (1991) 36-56.
15. Teorey, T., Yang, D., Fry, J.: A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model, *ACM Computing Surveys*, 18 (2), June, 1986.

16. Zhuge, H.: Resource Space Grid: Model, Method and Platform, *Concurrency and Computation: Practice and Experience*, 16 (14) (2004) 1385-1413
17. Zhuge, H.: The Knowledge Grid, World Scientific, Singapore (2004)
18. Zhuge, H.: Resource Space Model, Its Design Method and Applications, *Journal of Systems and Software*, 72 (1) (2004) 71-81
19. Zhuge, H., Xing, Y.: Integrity Theory for Resource Space Model and Its Application, Keynote, *WAIM2005*, LNCS 3739, (2005) 8-24
20. Zhuge, H., Yao, E., Xing, Y., Liu, J.: Extended Normal Form Theory of Resource Space Model, *Future Generation Computer Systems*, 21 (1) (2005) 189-198.
21. Zhuge, H.: The Open and Autonomous Interconnection Semantics, Keynote at 8th International Conference on Electronic Commerce, August 14-16, 2006, Fredericton, New Brunswick, Canada.

Next Generation Semantic Web Applications

Enrico Motta and Marta Sabou

Knowledge Media Institute
The Open University, UK
{e.motta, r.m.sabou}@open.ac.uk

Abstract. In this short paper, we examine current Semantic Web application and we highlight what we see as a shift away from first generation Semantic Web applications, towards a new generation of applications, designed to exploit the large amounts of heterogeneous semantic markup, which are increasingly becoming available. Our analysis aims both to highlight the main features that can be used to compare and contrast current Semantic Web applications, as well as providing an initial blueprint for characterizing the nature of Semantic Web applications. Indeed, our ultimate goal is to specify a number of criteria, which Semantic Web applications ought to satisfy, if we want to move away from conventional semantic systems and develop a *new generation of Semantic Web applications*, which can succeed in applying semantic technology to the challenging context provided by the World-Wide-Web.

1 Introduction

The past few years have witnessed a growing interest in the Semantic Web [1], as shown by the rapid increase of the amount of semantic markup available on the Web, by the growing number of organizations starting research and development activities in this research area, and by the number of Semantic Web applications, which now exist. Indeed, current data appear to show that the growth of the Semantic Web is mirroring that of the Web in the early nineties [2], a strong indicator that a large scale Semantic Web is likely to become a reality sooner rather than later.

The availability of semantic markup opens up novel possibilities to develop smart, web-based functionalities. For instance, in the brief history of the Semantic Web we have already seen Semantic Web applications that support intelligent data aggregation and presentation [3, 4], semantic search [5], automatic annotation [6, 7], question answering [8], and Semantic Web browsing [9]. However, if we look closely at the way these applications make use of web-based semantic and non-semantic resources, we can highlight a clear distinction between those applications which truly embrace the Semantic Web paradigm, and those which are more akin to conventional knowledge-based systems. At a coarse-grained level of abstraction, this distinction can be expressed by the difference between 'closed' semantic systems, which typically use a single ontology to perform data aggregation in a domain-specific fashion, and 'open' systems, which are heterogeneous with respect to both the ontological characterization and the provenance of the semantic data they handle.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 24 – 29, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In this paper we will explore this issue in some detail and we will propose a set of features that, in our view, will increasingly characterize the Semantic Web applications. Our analysis aims to be both *descriptive* and *prescriptive*. Descriptively, the objective here is to characterize the space of current Semantic Web applications, provide dimensions to compare and contrast them, and identify key trends. Prescriptively, our goal is to specify a number of criteria, which Semantic Web applications ought to satisfy, if we want to move away from conventional semantic systems and develop a *new generation of Semantic Web applications*, which can succeed in applying semantic technology to the challenging context provided by the World-Wide-Web.

2 Features of Open Semantic Web Applications

In what follows we introduce seven dimensions for analyzing Semantic Web applications and we use them to characterize a representative sample of Semantic Web systems. In particular we will compare and contrast systems such as CS Aktive Space [3], which can be characterized as *first generation* Semantic Web applications, from more recent systems, such as PiggyBank [9], a Semantic Web browser, or PowerAqua [8], a question answering system for the Semantic Web, which in our view provide early examples of the *next generation* of Semantic Web applications.

- **Semantic data generation vs reuse.** Early Semantic Web applications, such as CS Aktive Space [3], were developed in a context in which little semantic information was available on the Web. Hence, these applications produced and consumed their own data, much like traditional knowledge-based applications. In contrast with CS Aktive Space, more recent applications, such as PiggyBank or PowerAqua, are designed to operate with the semantic data that already exist. In other words, they worry less about bootstrapping a Semantic Web, than about providing mechanisms to exploit available semantic markup[1].
- **Single-ontology vs multi-ontology systems.** A first generation system, such as CS Aktive Space, makes use of a specific ontology, in this case the AKT Reference Ontology [10], to support data aggregation and provide a unified semantic model to the data acquired from several different sources. In contrast with CS Aktive Space, neither PiggyBank nor PowerAqua rely on any specific ontology. On the contrary, these systems can consume any number of ontologies at the same time. The rationale for this choice is that these systems simply assume that they operate on a large scale Semantic Web, characterized by huge amounts of heterogeneous data, which could be defined in terms of many different ontologies. In this context, it clearly does not make much sense to make a ‘closed domain’ assumption. It is interesting here to compare PowerAqua with an earlier question answering system, AquaLog [11]. While AquaLog is also ontology-independent, it cannot use multiple ontologies concurrently to answer a particular query. In other words, while AquaLog also assumes that a Semantic Web query system must be able to

[1] Actually PiggyBank also provides bootstrapping mechanisms to extract semantic data from HTML, however these are meant to provide extra flexibility to the system, rather than being an essential aspect of its modus operandi.

operate with different ontologies, it still assumes that it is feasible to make use of one ontology at the time. This feature makes AquaLog especially suitable for semantic organizational intranets, where organizational data are usually annotated in terms of a single organizational ontology. However it clearly makes it unsuitable for the Semantic Web at scale, where heterogeneous semantic data may need to be combined to answer specific queries.

- **Openness with respect to semantic resources.** This criterion distinguishes between those systems which are closed with respect to the semantic data they use and those which are able to make use of additional, heterogeneous semantic data, at the request of their user. For instance, a system such as CS Aktive Space cannot take into account RDF data available from a particular Web site, in response to a request from a user who wish to use them. CS Aktive Space can only use the data that the system developers have scraped from the various relevant sites and re-described in terms of the AKT Reference Ontology. In contrast with CS Aktive Space, a system such as PowerAqua has no such limitation: if new data become available, PowerAqua can use them with no configuration effort, to try and answer queries.
- **Scale as important as data quality.** The key feature of the Web is its size. Because publishing on the Web is so inexpensive, it has grown “like a virus”, acquired gigantic proportions in an incredibly short time, and revolutionized the way we access and publish information, shop, operate our businesses, socialize, and interact with our peers and with organizations. As argued earlier, it is likely that the Semantic Web will follow a similar growth pattern and as a result we will soon be able to build applications, which will explore, integrate, and exploit large amounts of heterogeneous semantic data, generated from a variety of distributed sources. Given this context it is interesting to distinguish between those applications that are designed to operate at scale and those which are more similar to the small-medium sized knowledge-based applications of the past. Indeed all the Semantic Web applications mentioned in our introduction take scale seriously. The difference is primarily between those applications like PowerAqua, which do not require any extra effort to bring in new sources, and those like CS Aktive Space, which require additional programming to bring in new information. In other words, scale per se is much less a useful discriminator between the first and second generation of Semantic Web applications, than a system’s ability to link its performance to the amount of semantic data existing on the Web. Two important implications arise from this emphasis on scale as one of the key features of Semantic Web applications. Firstly, the moment a system has to reason with very large amounts of heterogeneous semantic data, drawn from different sources, then necessarily these systems have to be prepared to accept variable data quality. Secondly, intelligence in these large-scale semantic systems becomes a side-effect of a system’s ability to operate with large amounts of data, rather than being primarily defined by their reasoning ability. This aspect strongly distinguishes Semantic Web applications from traditional knowledge-based systems, where the intelligence of a system was typically defined in terms of its ability to carry out complex tasks, such as, for instance, diagnosis, planning, or scheduling [12].

- **Openness with respect to Web (non-semantic) resources.** In our view a system that operates on a large-scale, rapidly evolving Semantic Web, should also take into account the high degree of change of the conventional Web. In particular, platforms such as TAP [5] and PiggyBank [9] provide facilities for integrating data acquisition mechanisms in their architecture, to facilitate the extraction of data from arbitrary sources on the Web. Analogously, automatic annotation systems such as KIM [7] and Magpie [6] can work on any Web page, although of course the quality of the annotation may degrade if the page in question does not reflect the current ontology used by these systems to drive automatic annotation.
- **Compliance with the Web 2.0 paradigm.** As pointed out by Tim O'Reilly, a key principle behind the success of the Web 2.0 paradigm is that of *Harnessing Collective Intelligence*. In other words many of today's most successful Web applications are based on massively distributed information publishing and annotation initiatives, such as Wikipedia[2], Flickr[3], etc. While it ought to be emphasized that a large scale Semantic Web will primarily be constructed by exploiting automatic data generation and integration mechanisms, it is also important to note that Semantic Web applications cannot ignore the lessons from the success of Web 2.0 applications and therefore they ought to embed Web 2.0 features. If this premise is correct, then there are at least two very important implications. The first one is that, like typical Web 2.0 applications, Semantic Web systems also need to provide mechanisms for users to add and annotate data. Indeed, at conferences such as the European Semantic Web Conference we have already seen the value of allowing distributed semantic annotation. However tools to support user annotation are still rather primitive, and better tools are badly needed. Another important aspect of integrating Web 2.0 principles into Semantic Web activities concerns the need to integrate artefacts such as folksonomies into Semantic Web applications. Although systems such as PiggyBank already provide tagging mechanisms, this is only a preliminary step. The next step will be to perform a deeper integration of folksonomies and ontologies. In our group we are examining the use of relation extraction mechanisms to achieve this goal.
- **Open to services.** Web services have revolutionized the Web, transforming it from a static information space to a dynamic data sharing infrastructure. In our view it is also essential that Semantic Web applications integrate web service technology in their architecture, and indeed we can already highlight a number of applications that do so. For instance, both TAP [5] and PiggyBank [9] seamlessly integrate scraping services into their data acquisition architectures. Another good example of the use of services is Magpie [6], which integrates services into its annotation mechanisms, by dynamically associating a highlighted item with all the services which are relevant to the type of the item in question. For example, when highlighting an instance of class 'researcher', Magpie could automatically retrieve all services it knows about, which make sense for a researcher. For instance, one such service could list all the projects that a researcher is involved in.

[2] http://en.wikipedia.org/

[3] http://www.flickr.com/

3 Conclusions

The main conclusion of the above analysis is that the growth of the Semantic Web has been promptly followed by changes in the way Semantic Web applications are developed. By analyzing and contrasting some older and newer systems, we have identified a set of features, which in our view will characterize the next generation of Semantic Web applications. In particular we observe that the latest Semantic Web systems are geared to take advantage of the vast amount of heterogeneous semantic data available online. Freed from the burden of creating their own semantic data, they concentrate on finding and meaningfully combining the available semantic markup. In our view this is not an accidental feature but an important indicator of a shift taking place from the first generation of Semantic Web systems to the next one. In a nutshell, next generation Semantic Web systems will necessarily have to deal with the increased heterogeneity of semantic sources.

Finally, Semantic Web applications will also tend to reflect the major developments in conventional Web systems, and as a result in the future we will see an increasing degree of integration of key Web technologies, such as social tagging and web services, in Semantic Web applications.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American*, 284(5):34 – 43. 2001.
2. Lee, J., Goodwin, R.: The Semantic Webscape: a View of the Semantic Web. In WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web, pages 1154--1155, New York, NY, USA, 2005. ACM Press.
3. Schraefel, M.C., Shadbolt, N.R., Gibbins, N., Glaser, H., Harris, S.: CS AKTive Space: Representing Computer Science in the Semantic Web. In Proceedings of the 13th International World Wide Web Conference.
4. Hyvönen E., Mäkelä E., Salminen M., Valo A., Viljanen K., Saarela S., Junnila M. and Kettula S.: MuseumFinland - Finnish Museums on the Semantic Web. Journal of Web Semantics, vol. 3, no. 2, pp. 25, 2005.
5. Guha, R. and McCool, R.: Tap: a semantic web platform. Computer Networks, 42(5):557--577, August 2003.
6. Dzbor, M., Motta, E., and Domingue, J.B.: Opening Up Magpie via Semantic Services. In McIlraith et al. (eds), The SemanticWeb - ISWC 2004, Third International Semantic Web Conference. Hiroshima, Japan, November 2004. Lecture Notes in Computer Science, 3298, Springer-Verlag 2004.
7. Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., and Goranov. M.: KIM – A Semantic Annotation Platform. In D. Fensel, K. Sycara, and J. Mylopoulos (eds.), The Semantic Web - ISWC 2003, Second International Semantic Web Conference. Lecture Notes in Computer Science, 2870, Springer-Verlag, 2003.
8. Lopez, V., Motta, E., Uren, V.: PowerAqua: Fishing thSemantic Web. In York Sure and John Domingue (eds.), The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006. Lecture Notes in Computer Science 4011, Springer 2006, ISBN 3-540-34544-2.

9. Huynh, D., Mazzocchi, S., Karger, D.: Piggy Bank: Experience the Semantic Web Inside Your Web Browser. In Gil et al. (eds), The Semantic Web - ISWC 2005, 4th International Semantic Web Conference, ISWC 2005. Galway, Ireland, November 6-10, 2005. Lecture Notes in Computer Science, 3729 Springer-Verlag, 2005.
10. AKT Reference Ontology. http://www.aktors.org/publications/ontology/.
11. Lopez, V., Motta, E.: Ontology Driven Question Answering in AquaLog. 9th International Conference on Applications of Natural Language to Information Systems (NLDB 2004).
12. Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W., and Wielinga, B. Knowledge Engineering and Management - The CommonKADS Methodology. MIT Press, December 1999.
13. Mika, P.: Flink: SemanticWeb Technology for the Extraction and Analysis of Social Networks. *Journal of Web Semantics*, 3(2), 2005.

Hierarchical Topic Term Extraction for Semantic Annotation in Chinese Bulletin Board System

Xiaoyuan Wu, Shen Huang, Jie Zhang, and Yong Yu

Shanghai Jiao Tong University No.800, Dongchuan Road, Shanghai, China 200240
{wuxy, huangshen, zhangjie, yyu}@sjtu.edu.cn

Abstract. With the current growing interest in the Semantic Web, the demand for ontological data has been on the verge of emergency. Currently many structured and semi-structured documents have been applied for ontology learning and annotation. However, most of the electronic documents on the web are plain-text, and these texts are still not well utilized for the Semantic Web. In this paper, we propose a novel method to automatically extract topic terms to generate a concept hierarchy from the data of Chinese Bulletin Board System (BBS), which is a collection of plain-text. In addition, our work provides the text source associated with the extracted concept as well, which could be a perfect fit for the semantic search application that makes a fusion of both formal and implicit semantics. The experimental results indicate that our method is effective and the extracted concept hierarchy is meaningful.

1 Introduction

The Semantic Web relies heavily on formal ontologies to structure data for comprehensive and transportable machine understanding. Hence, the Semantic Web's success and proliferation depends on quickly and cheaply constructing domain-specific ontologies. Manual ontology acquisition remains a tedious, cumbersome task that can easily result in a knowledge acquisition bottleneck. Ontology learning could greatly help ontology engineers construct ontologies. However, the spread usage of ontology learning is still mainly focused on the structure data like database and XML with definite schema etc, as well as some semi-structure data, like dictionaries. However, there are seas of web pages in the Internet and nearly all of them contain free texts in natural language, and these texts are still not well utilized for the Semantic. In this paper, we propose techniques of topic term extraction to generate a concept hierarchy from a collection of documents, which could enrich the result of concept structure construction from the original plain-text.

The plain-text in Chinese BBS dataset is the focus of our research work. Bulletin Board System (BBS) is a kind of web virtual space where people can freely discuss anything. Especially in China, people always have great excitement for participating in the activities in BBS. Besides, some famous websites like sina.com and sohu.com also have their own BBS portals, which attract many users taking part in. According to the statistics published by China Internet Network Information Center in 2004, the scale of virtual community like BBS in China has expanded to 27.6% of the whole

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 30–43, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Chinese Web[1]. Thus, the power of "word of mouth" gradually appears and should not be ignored. Nevertheless, to our best knowledge, few studies were focused on the BBS data for the Semantic Web.

Ideal methods for search in semantic knowledge base should rely on logical reasoning which is formal and precise. However, due to the gap between the traditional web and semantic web, current methods are generally a combination of the formal semantics and traditional IR techniques, e.g., our previous work on semantic search [22][10]. However, lack of data is a great limit for those searching models, since they need the data both with ontology and texts associated with ontological concept. In this paper, we propose a method to automatically extract ontology, more precisely concept hierarchy, from each board of BBS portals. Thus, combined with the original texts, BBS portals could provide the appropriate type of datasets for the semantic searching models.

Our research is mainly concerned with designing algorithms to automatically extract topic terms from thousands of BBS messages to generate a concept hierarchy. Our method consists of two main components. First, we take well advantage of special features of BBS to extract parent topic terms. Those features are combined together by regression models, and the final score is used to determine whether a term is important or not. In addition, each parent topic term could represent one subset of the whole corpus. Second, we extract child topic terms for each parent term. We utilized a mixture model of subsuming probability and co-occurrence to determine if the two terms have a real parent-child relationship. We show a part of concept hierarchy of the "anti-virus" board in Figure 1. The data presentation is enriched by our method rather than thousands of messages with plain-text. Besides, the results will be useful in applications of the Semantic Web.

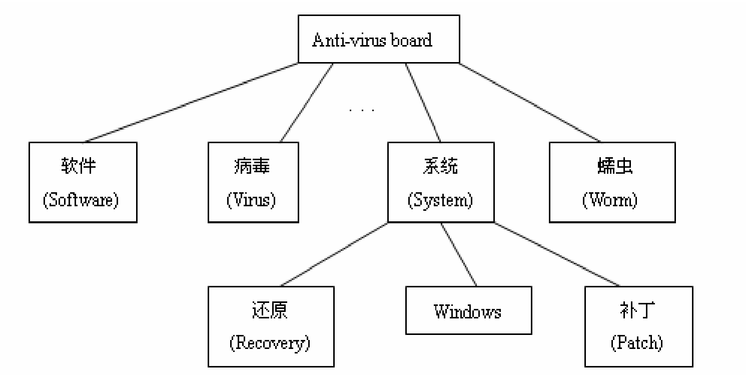


Fig. 1. The concept hierarchy of the Anti-virus board

The remainder of paper is organized as follows: In Section 2, we explain our topic term extraction algorithms. In Section 3, we discuss the concept hierarchy. In Section 4, we introduce the experimental methods and analyze the experimental results. In Section 5, we briefly describe some related work. In Section 6, we conclude our work and discuss the future work.

[1] http://tech.sina.com.cn/focus/04_net_rep/index.shtml

2 Algorithms of Hierarchical Topic Term Extraction

The goal of our proposed method is to generate a concept hierarchy from a collection of free texts. Our technique is mainly composed of three steps:

1. Parent topic term extracting and ranking,
2. Child topic term extracting and ranking,
3. Grouping synonyms.

In this section, we will discuss these steps in detail.

2.1 Parent Topic Term Extraction and Ranking

The purpose of this step is to identify topic terms from an initial set of messages. First, we consider some particular features of terms in Chinese BBS messages. Second, we briefly introduce how to use regression models to calculate the score of each candidate term. Third, we present a co-occurrence algorithm to refine the results of topic term extraction. Finally, we will tackle with the problem of synonym.

- **Feature Extracting**

We list four features calculated during messages preprocessing. These features are supposed to be relative to the scores of terms.

TF.IDF

Intuitively, more frequent terms are more likely to be better candidates of topic terms. The motivation for usage of an IDF factor is that terms which appear in many messages are not useful for representing a sub-topic.

$$TFIDF = f(t) \cdot \log \frac{N}{|D(t)|} \tag{1}$$

where *f*(*t*) denotes the frequency of term *t* and *D*(*t*) represents the set of messages that contain term *t*. Besides, we denote *N* as the number of messages.

Term Frequency in Message Title

When users post a message to BBS, they usually describe their questions or opinions briefly in title. We calculated the term frequency in title (denoted by *TITLE*).

Average of Term's First Occurrence in Message

First occurrence is calculated as the number of terms that precede the term's first appearance, divided by the number of terms in a message. The result is a number between 0 and 1 that represents how much of the message precedes the term's first appearance. Topic terms are relative important and assumed to be appeared in the front of messages. We use *ATFO* to denote this property.

$$ATFO = 1 - \frac{1}{|D(t)|} \cdot \sum_{d \in D(t)} \frac{p(td)}{\#d} \tag{2}$$

where *p*(*td*) denotes the number of terms that precede the term *t*'s first appearance in message *d* and *#d* represents the number of terms in message *d*. Terms with higher *ATFO* value are preferred.

Message Depth in a Thread

It is a common phenomenon that there are many topic drift phenomena in BBS portals. As the message number in a thread becomes larger, the topic is more likely to drift away from the original one. It is assumed that terms appearing in first several messages of a thread should be less influenced by noisy data. The average value of message depth can be calculated by,

$$DEPTH_t = \frac{1}{f(t)} \sum_{d \in D(t)} [Depth(d) \cdot f(td)] \tag{3}$$

For example, If a message is the root of a thread, then *Depth* = 1, and if a message follows a root message, then *Depth* = 2.

- **Learning to Rank Topic Terms**

Given the above four properties, we could use a single formula to combine them and calculate a single salience score for each term. Thus, we utilize training data to learn a regression model. In this paper, we use linear regression and support vector regression to complete this task. These two models are both widely used in regression.

Linear Regression: The relationship between two variables is modeled by fitting a linear equation to observed data. The linear regression model postulates that:

$$y = b_0 + \sum_{j=1}^{p} b_j x_j + e \tag{4}$$

In our case, independent variable x = (*TFIDF*, *TITLE*, *ATFO*, *DEPTH*), and dependent y can be any real-valued score. We use y to rank topic terms in a descending order, thus the topic terms could be ranked by their importance.

Support Vector Regression: In support vector regression, the input x is first mapped onto a high dimensional feature space using some nonlinear mapping, and then a linear model is constructed in this feature space. Support vector regression uses a new type of loss function called ε-insensitive loss function:

$$L_\varepsilon(y, f(x,\omega)) = \begin{cases} 0 & \text{if} |y - f(x,\omega)| \le \varepsilon \\ | y - f(x,\omega) | - \varepsilon & \text{otherwise} \end{cases} \tag{5}$$

Support vector regression tries to minimize $\|\omega\|^2$. This can be described by introducing (non-negative) slack variables ξ_i, ξ_i^*, i=1,..., n, to measure the deviation of training samples outside ε-insensitive zone. Thus support vector regression is formalized as minimization of the following functional:

$$\min \frac{1}{2} \|\omega\|^2 + C\sum_{i=1}^{n} \xi_i + C\sum_{i=1}^{n} \xi_i^* \tag{6}$$

$$\begin{aligned} & y_i - f(x_i, \omega) \le \varepsilon + \xi_i^* \\ & f(x_i - \omega) - y_i \le \varepsilon + \xi_i \\ & \xi_i^*, \xi_i \ge 0, i = 1,\dots,n \end{aligned} \tag{7}$$

- **Topic Term Extraction Refinement**

In this step, we want to refine the results of our topic term extraction. As we mentioned before, we combine four features to determine the score of each term. Some terms such as, “问题”(question), “大虾”, “高手”(these two words are not regular Chinese words, but frequently used on web), are wrongly considered as topic terms in our corpus. The reason is that Question/Answer plays a significant role in daily activities in Chinese BBS, and such words appear frequently both in message body and title.

However, these words are non-informative and are not appropriate to represent subtopic of messages.

We make use of co-occurrence information to refine our topic term extraction algorithm. In Figure 2, we list some neighbor terms of "问题"(question) and "病毒"(virus). One is a fake topic term and the other is a true topic term. The number behind each neighbor term is the frequency of the terms and their neighbors appear together in a fixed window. Numbers are not integral because we give different weights to neighbor terms by their distances from the candidate topic term.

问题(question):
解决 (solve),46.8;出现 (appear),33.8; 时间 (time), 22.4;遇到 (meet),22.0; 没有 (no),21.6 …

病毒(virus):
软件(software),71.2 ;文件(file),65.4; 木马(horse),60.5; 发现 (discover)54.6; 扫描 (scan), 46.6…

Fig. 2. Several neighbor words of "问题'(question) and "病毒'

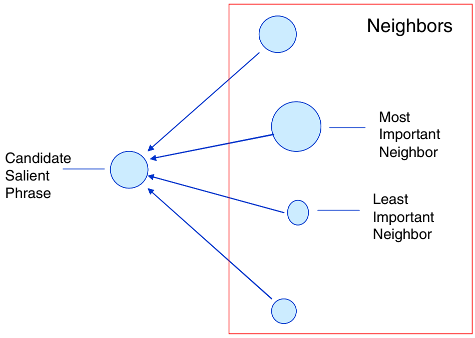


Fig. 3. Relationship between neighbors and a candidate topic term

In Figure 3, we present a relationship between each term and its neighbors. The left circle represents a term and the right four circles represent its neighbor terms. The bigger circle means the term is more important, namely has higher score. We could observe from Figure 2 that the neighbors of a fake topic term like "问题'(question) are almost all poor-quality terms, while the neighbor of a true topic term like "病毒'(virus) are more informative. Therefore, we could make use of the quality of neighbor terms and co-occurrence weight to calculate the new score of each candidate topic term.

$$RefinedScore_i = \sum_{j \in N(i)} Score_j \times Weight_{ij} \tag{8}$$

where $N(i)$ represents the set of neighbors of term i. *Score* is calculated by the regression model and *Weight* means frequency of a candidate topic term and its neighbors appear together. Finally, we get the final score of each term i,

$$FinalScore_i = RefinedScore_i + Score_i \tag{9}$$

When terms ranked by this final score, we could get rid of some terms like "问 题'(question), "大 虾' and "高 手'.

2.2 Finding Child Topic Terms

In order to extract child topic terms for each parent topic term, we introduce a method presented in [18]. It is defined as follows, for two terms, x and y, x is said to subsume y if the following conditions hold, $P(x|y) \geqslant 0.8$, $P(y|x) < 1$. In other words, x subsumes y if the documents which y occurs in are a subset of the documents which x occurs in. In the hierarchy of term, x is the parent of y.

We calculate $P(x|y)$ to get top n terms with high probabilities as the child topic term of x. However, the result is not satisfied. The terms ranked high by this method are truly subsumed by parent terms, but some of those terms are non-informative to users. For example, some rarely appeared terms are subsumed by parent terms, but they can not represent sub-topics. Therefore, even if the value of $P(x|y)$ equals 1, some terms are still not good candidates for child topic term.

To solve this problem, we extract child topic terms from the neighbor terms of parent topic terms. For example, "设 置'(configure) often occurs nearby "防 火 墙' (firewall), thus "设置'(configure) seems to be a good child term and represent an aspect of "防 火 墙'(firewall). One advantage that we restrict the child topic term candidates in the range of neighbor terms is to reduce noise in results. The simple heuristic seems to work well in practice. Therefore we use following two features together to determine the score of each candidate term.

1. $P(x|y)$, the probability of x in the condition of y occurs
2. Frequency of term x and y's co-occurrence in a small size window.

Thus, we use a linear combination formula to determine the score of each candidate of child term. The value of $P(x|y)$ and CoOccurrence are both normalized.

$$ChildScore = \lambda \times P(x \mid y) + (1-\lambda) \times CoOccurrence \qquad (10)$$

where y is a term in neighborhood of parent term x.

If we hope to find more levels of topic terms, we can run this method recursively.

2.3 Grouping Synonyms

It is common that people use different words to describe the same concept. In our Chinese BBS dataset, we found three basic types of synonyms. First, many users often use abbreviation like "卡巴' for "卡巴斯基' (Kaspersky). Second, miss spelling, like "特洛伊'(Trojan) and "特洛依'. Finally, some regular synonyms, such as "配 置' and "设 置' which both mean "configure". Some papers [16][20] solve this synonym problems by WordNet [3]. However, to our best knowledge, there are no similar tools available to identify Chinese synonyms. To solve this problem, we use Levenshtein Distance (LD) algorithm [14] and mutual information to group Chinese synonyms.

- **Levenshtein Distance**

Levenshtein Distance is a measure of the similarity between two strings. The distance is the number of deletions, insertions, or substitutions required to transform into. For example, if S_i is "特洛伊' and S_j is "特洛依', then LD(S_i, S_j) = 1, because one substitution (change "伊" to "依") is needed. The smaller the Levenshtein Distance is, the more similar the strings are. This can be applied to the identification of Chinese synonyms.

- **Mutual Information**

The Mutual Information of two random variables is a quantity that measures the independence of the two variables, as in equation (11),

$$I(S_i, S_j) = \log \frac{P(S_i, S_j)}{P(S_j)P(S_j)} \tag{11}$$

$P(S_i,S_j)$ is the joint probability of the term S_i and S_j. Its maximally likelihood estimator is n_{ij}/N, where n is the number of messages involving both S_i and S_j, and N is the number of total messages. $P(S_i)$ and $P(S_j)$ are probabilities of the terms S_i and S_j, which can be estimated as n_i/N and n_j/N respectively. If S_i and S_j are independent, $I(S_i,S_j)$ is zero. However, if S_i and S_j often co-occur in the same messages, $I(S_i,S_j)$ will turn out to be high.

Finally, the above two measures are combined together to judge whether two phrases should be grouped.

$$SM_{ij} = \beta \cdot \frac{Len_i + Len_j}{2LD_{ij}} + (1-\beta) \cdot I(S_i, S_j) \tag{12}$$

where Len_i and Len_j are the length of S_i, S_j. According to our previous work in [13], β is set to 0.6. In addition, by trial and error, S_i and S_j are considered as synonyms if $SM_{ij} \geq 3$.

3 Concept Hierarchy of BBS Boards

The conceptual structures that define an underlying ontology provide the key to machine processable data on the Semantic Web. Ontologies serve as metadata schemas, providing a controlled vocabulary of concepts, each with explicitly defined and machine processable semantics. The integration of knowledge acquisition with machine learning techniques proved extremely beneficial for knowledge acquisition. The drawback to such approaches, however, was their rather strong focus on structured knowledge or databases, from which they induced their rules. Besides, current methods are generally a combination of the formal semantic and traditional IR techniques, because of the gap between the traditional web and semantic web. However, lack of data is a great limit for those searching models, since they need the data both with ontology and texts associated with ontological concept.

The effort behind the Semantic Web is to add semantic annotation to Web documents in order to access knowledge instead of unstructured material, allowing knowledge to be managed in an automatic way. Web mining can help to learn definitions of structures for knowledge organization (e. g., ontologies) and to provide the population of such knowledge structures [1]. In this paper, we also make use of web mining techniques for the Semantic Web. Concept hierarchies of BBS boards provide the text source associated with the extracted concept. Combined with the original texts, BBS portals could provide the appropriate type of datasets for the semantic applications.

Our concept hierarchy could assist the knowledge engineer in extracting the semantics, but cannot completely replace her. In order to obtain high-quality results, one cannot replace the human in the loop, as there is always a lot of tacit knowledge involved in the modeling process. A computer will never be able to fully consider background knowledge, experience, or social conventions. The overall aim of our research is thus not to replace the human, but rather to provide him with more support.

4 Experiments

In this section, we will introduce several experiments to prove the effectiveness of the proposed methods. First, we describe the experiment setup and evaluation of parent topic terms extraction. Second, child topic terms extraction and parent-child relationship are both closely examined.

4.1 Parent Topic Term Extraction

All the experiments are based on 6,458 messages crawled from the websites in Table 1. The main topic of this dataset is about "computer virus".

Table 1. Websites of our messages resource

.com.cn	pconline.com.cn, zol.com.cn, zdnet.com.cn, enet.com.cn
.com	forum.ikaka.com, chinadforce.com, qq.com, yesky.com
.net	pchome.net, langfang.net

4.1.1 Evaluation Measure

We use precision at top N results to measure the performance:

$$P@N = \frac{|C \cap R|}{R} \tag{13}$$

where R is the set of top N topic terms extracted by our method, and C is the set of manually tagged correct topic terms. For parent topic term, we use $P@10$, $P@20$ and $P@30$ for evaluation.

4.1.2 Training Data Collection

After all the messages are preprocessed, we get a list of candidate topic terms which are all nouns or nouns phrases by a Chinese POS tagger[7]. We first simply rank terms by their term frequency, and top 200 terms are used for labeling, since the term number of the whole corpus is too large and the terms appearing rarely are hardly representative for topics. Four graduate students are invited to label these terms. We assign 1 to y values for terms that are considered as topic, and assign 0 to y values for others. These y values together with term features are used in regression.

4.1.3 Experimental Result

We first use the each single feature described in Section 2.1 to rank terms. The average precision at top 10, 20 and 30 are shown in Figure 4.

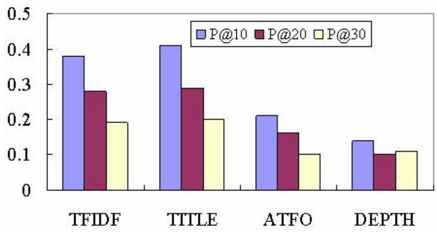


Fig. 4. Performance of each single property

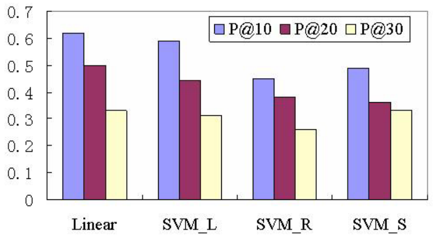


Fig. 5. Performance of each regression model

From Figure 4, *TFIDF* and *TITLE* are much better indictors for topic terms than other features. We prove our hypothesis that in BBS, users usually put the most important information in titles to attract others' attention. *ATFO* does not work well in our dataset, since our messages not only contain pure Question\Answer and opinion discussion messages, but also include some long advertisements and long documents transcribed from other websites. These two are common phenomena in Chinese BBS portals. Besides, depth of messages in threads (*DEPTH*) is also not a good indicator in this dataset. The reason might be that our "computer virus" dataset is technique oriented, and there are less topic drift phenomena existed. Maybe, in some boards like "news" and "entertainment", depth will be an important feature.

In order to do regression analysis, we partition the dataset into four parts and use four-fold cross validation to evaluate the average performance of topic term extraction. For support vector regression, three kernel functions are used here: linear kernel (denoted by SVM_L), RBF kernel (denoted by SVM_R) and sigmoid tanh kernel (denoted by SVM_S). The comparison of these regression models is shown in Figure 5. The precision achieved by linear regression and SVM_L are almost same and both gain significant improvements than each single feature in Figure 4.

In Section 2.1, we introduced our refinement method to further remove some non-informative terms. In Table 2, we compare the original result with the refined one. We can see that some poor-quality words like "问题'(question), "高手' and "大虾' are wrongly extracted to be topic words because of their high frequencies both in title and text. After we use the neighbor words to refine the original result, we find these words are all removed. Moreover, two more informative words about "computer virus", "蠕虫' (worm) and "卡巴斯基' (Kaspersky), are discovered. In short, the new result seems to be more reasonable.

Table 2. Topic terms extraction comparison

No Refined	病毒(virus) 软件(software) 瑞星(Rising) 诺顿(Norton) 防火墙(firewall) **问题(question)** 木马(horse) 电脑(computer) 文件(file) 鸽子(pigeon) 系统(system) 高手 金山(Kingsoft) 日志(log) 网络(network) 方法(method) 进程(computer process) 垃圾(rubbish) 大虾网页(web page)
Refined	软件(software) 病毒(virus) 文件(file) 木马(horse) 瑞星(Rising) 防火墙(firewall) 系统(system) 程序(program) 诺顿(Norton) **蠕虫(worm)** 工具(tool) 金山(Kingsoft) 网络(network) 进程(computer process) 用户(user) 功能(function) 电脑(computer) **卡巴斯基(Kaspersky)** 个人(personal)

We hope to generate a concept hierarchy to represent the original free texts. Thus, the topic terms extracted by our methods should cover the main topics of the corpus. Meanwhile the overlap between two topic terms should be as small as possible. Figure 6 shows the message coverage when topic terms are extracted. The X-axis indicates the top 10 topic terms and Y-axis is the percent of coverage. We could observe that the 10 topic terms could represent nearly 60% of the whole messages. Besides, Figure 7 shows the overlap of the same top 10 topic terms. For example, the message overlap of top 5 topic terms is about 30%, which means there are 70 distinct messages in 100 messages. In the future, we will further refine the term extraction algorithm by trying to maximize the message coverage while at the same time minimizing the message overlap.

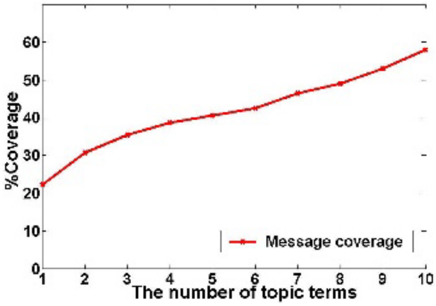


Fig. 6. Message coverage of extracted topic terms

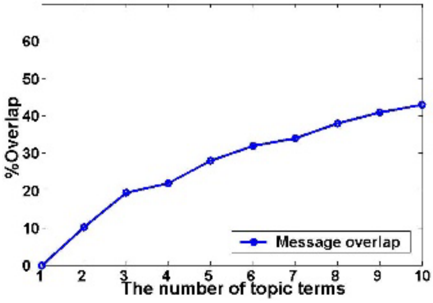


Fig. 7. Message overlap of extracted topic terms

4.2 Child Topic Term Extraction

- **Experiment Setup**

In this part, child topic term extraction method will be evaluated. Evaluating the concept hierarchies is really a challenge. In paper [18], the authors designed experiments to label the relationship between parent terms and child terms, and judged if there was a parent-child relationship existed. We also use this method to evaluate our results. Four graduate students are asked to decide on the type of relationship between child and parent. Five of the organizing relations in [18] were presented here.

1. Child is an aspect of parent, e.g. “端 口”(port) is an aspect of “防火墙”(firewall).
2. Child is a type of parent, e.g. “特洛伊”(Trojan) is a type of “木马”(horse).
3. Child is as same as parent.
4. Child is opposite of parent.
5. Do not know or they have some other relations.

The first two relation types indicate that a child is more specific than its parent, namely this parent-child pair is meaningful. We are mainly concerned with the most important topic terms, so top 10 parents each with its top 10 child topic terms are judged according to the five types of relations above.

- **Experimental Result**

We use $P(x|y)$ and co-occurrence frequencies together to calculate the scores of candidate child topic terms. The method has been described in Section 2.2. We set the default neighbor size as 10, namely we choose five terms before and five terms after parent term to form the pool of child terms candidate and calculate the co-occurrence frequency. In addition, we set λas 0.7 in formula (8) for the experiment using some heuristics.

As it can be seen in Table 3, when we combine two features together, we get 69% (51% + 21%) of the parent-child pairs have the "aspect" or "type" relationships, which outperforms that using the two features separately. By observing the data, we found two reasons. First, if we use $P(x|y)$ only, the ratio of "don't know" is highest in that column. It is very common that some low frequencies terms can get high score in $P(x|y)$. For example, "雨燕'(name of an author) has very low word frequency, but all the occurrences are together with the term "诺 顿'(Norton). Thus, the score of $P(x|y)$=1, and "雨 燕'(name of an author) is wrongly selected as a child term of "诺顿'(Norton). Second, if we use co-occurrence only, we get the lowest value in the column of "type". Many parent-child pairs have real good "type" relationships, but the frequencies of co-occurrence are not high enough to get high scores. For example, "木 马'(horse) and "特 洛 伊'(Trojan), "蠕 虫' (worm) and "MyDoom"(a kind of worm), etc are all discarded because of their lower frequencies. However, these pairs are true child topic terms.

Table 3. Comparison of different extraction methods

	aspect	type	same	opposite	don't know
$P(x\|y)$	23%	25%	7%	1%	44%
Co-occurrence	38%	15%	9%	0%	38%
$P(x\|y)$ & Co-occurrence	51%	21%	7%	1%	20%

In Table 4, we show four groups of results when we use different neighbor sizes. We want to check if the neighbor size will affect the effectiveness of this method. As Table 4 shows, the performance differences are tiny. Intuitively, the bigger the size is, the more candidate terms return. However, in our experiment, the changed sizes almost have no influences on top 10 child terms.

Table 4. Comparison of different neighbor sizes

	aspect	type	same	opposite	don't know
Size=4	44%	25%	7%	0%	26%
Size=6	39%	27%	6%	1%	37%
Size=8	42%	23%	7%	0%	28%
Size=10	51%	21%	7%	1%	20%

By applying the child topic term extraction method recursively, we could obtain a concept hierarchy from a collection of free texts. Because of the limited space here, in Table 5, we list top 10 parent topic terms and their top 5 child topic terms. From Table 5, we could observe that the result is not good enough to replace human. However, we assist engineers a lot, and a few efforts by human intervention based on our extracted hierarchy could achieve better results.

Table 5. Topic term extraction results

Parent Topic Terms	Child Topic Terms
1. 软件 (software)	1. 间谍(spy) 2. 名称(name) 3. 防毒软件(anti-virus software) 4. 安装软件(install software) 5. 应用软件(application software)
2. 病毒(virus)	1. 发现病毒(discover virus) 2. 名称(name) 3. 样本(sample) 4. 病毒扫描(virus scan) 5. 更新病毒(update virus)
3. 文件(file)	1. 删除文件(delete file) 2. 隐藏文件(hiding file)3. 压缩文件(zip file) 4. 备份(backup) 5. exe
4. 木马(horse)	1. 清除木马(delete horse) 2. 克星(adversary) 3. 木马清除(horse delete) 4. 防线(line of defense) 5. 特洛伊(Trojan)
5. 瑞星(Rising, a Chinese anti-virus company)	1. 江民(a Chinese anti-virus company) 2. 瑞星升级(Rising upgrade) 3. 使用瑞星(using Rising) 4. 网站(website) 5. 安装瑞星(install Rising)
6. 防火墙(firewall)	1. 安装防火墙(install firewall) 2. 防火墙设置(firewall configure) 3. 防火墙保护(firewall protection) 4. 穿过防火墙(get through) 5. 过滤防火墙(filtering firewall)
7. 系统(system)	1.还原(recovery) 2. 资源(resource) 3. Windows 4. 关闭系统(shutdown system) 5. 补丁(patch)
8. 程序(program)	1. 应用程序(application program) 2. 安装程序(install program) 3. 运行程序(run program) 4. 驱动程序(device driver) 5. 恶意(hostility)
9. 诺顿(Norton)	1. 诺顿升级(Norton upgrade) 2. 诺顿提示(Norton reminding) 3. 暂停诺顿(pause Norton) 4. WinXp 5. 江民(a Chinese anti-virus company)
10. 蠕虫(worm)	1. IRC 2. 传播蠕虫(spread worm) 3. MyDoom 4. Whitty 5. 副本(copy)

5 Related Work

Significant amount of research on Information Extraction (IE) has been performed in various projects (e.g., [1][3][15]). They provided tools such as tokenizers, part-of-speech taggers, gazetteer lookup components, pattern-matching grammars, coreference resolution tools and others that aid the construction of various NLP and especially IE applications.

A crucial aspect of creating the Semantic Web is to enable users to create machine readable web content. Emphasis in the research community has till now been put on building tools for manual annotation of documents (e.g. [6]). Recently, some studies focused on producing automatic or semi-automatic methods for annotating documents, such as [4][11]. Mori [16] proposed a key extraction method to automatically annotate personal metadata. The work in [14] proved that web mining can help to build the Semantic Web.

The problem of keyword extraction (topic term) has been investigated in a number of studies, e.g. [14][20][21]. Most of them focused on extracting keywords from a single document, while our method is to extract keywords from a corpus.

Some topic finding studies (e.g. [18][8]) are related to our method. Sanderson and Croft [18] built concept hierarchies by finding pairs of concepts (x, y) in which x subsumes y. Lawrie and Croft [8] used the Dominating Set Problem for graphs to choose topic terms by considering their relation to the rest of the vocabulary used in the document set.

Keyword extraction also has been used in some previous studies about clustering search results. The work in [23] extracted topic words from web search results, and then clustered results using these topic words. Our work is related but quite different because we use special features of BBS message to rank each candidate topic term and extract child topic terms to get more specific topics and form a concept hierarchy.

6 Conclusion and Future Work

With the current growing interest in the Semantic Web, the demand for ontological data has been on the verge of emergency. However, most of the electronic documents on the web are plain-text, and these texts are still not well utilized for the Semantic Web. In this paper, we propose a novel method to automatically extract topic terms to generate a concept hierarchy from the data of Chinese Bulletin Board System (BBS), which is a collection of plain-text. In addition, our work provides the text source associated with the extracted concept as well, which could be a perfect fit for the semantic search application which makes a fusion of both formal and implicit semantics. Several special features of BBS and regression models are utilized to extract parent topic terms. Beside, child topic terms could be extracted recursively by a mixture model of subsuming probability and co-occurrence. The effectiveness has been verified by our experiment results.

As the future work, we plan to study the issues of (1) relations extraction and annotation from BBS boards to form a more integrated ontology and (2) exploring the formal and general approach of concept hierarchy evaluation and validation.

References

1. Berendt, B., Hotho, A., and Stumme, G. Towards semantic web mining. In International Semantic Web Conference (ISWC02), 2002
2. Cunningham, H. Information Extraction: a User Guide (revised version). Department of Computer Science, University of Sheffield, May, 1999.
3. Cunningham, H., Maynard, D., Bontcheva K. and Tablan V., GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics, 2002.
4. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J. A., and Zien, J. Y. 2003. Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In The Twelfth International World Wide Web Conference (WWW2003).

5. Fellbaum, C. WordNet: on Electronic lexical Database, MIT Press.
6. Handschuh, S., Staab, S., and Maedche, A. Creating relational metadata with a component-based, ontology driven frame work. In proceeding sofK-Cap2001 (Victoria, BC, Canada, October 2001).
7. http://www.nlp.org.cn
8. Lawrie D. and Croft W. B. Finding Topic Words for Hierarchical Summarization. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01).
9. Levenshtein, V.I. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Cybernetics and Control Theory 10 (1966) 707-710.
10. Li, L., Liu, Q.L., Zhang, L. and Yu, Y. PDLP: Providing an Uncertainty Reasoning Service for Semantic Web Application in Proc. of the Eighth Asia Pacific Web Conference (APWeb2006).
11. Li, Y., Zhang, L., and Yu, Y. Learning to Generate Semantic Annotation for Domain Specific Sentences. In: K-CAP 2001 Workshop on Knowledge Markup & Semantic Annotation, October 21, 2001, Victoria B.C., Canada.
12. Liu, B., Hu, M., and Cheng, J.H. Opinion Observer: Analyzing and Comparing Opinions on the Web. WWW 2005, Chiba, Japan.
13. Liu, W., Xue, G.R., Huang, S. and Yu, Y. Interactive Chinese Search Results Clustering for Personalization. The 6th International Conference on Web-Age Information Management (WAIM2005), Hangzhou, China, October 11-13, 2005.
14. Matsuo, Y., and Ishizuka, M. Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information. International Journal on Artificial Intelligence Tools.
15. Maynard, D., Tablan, V., Bontcheva, K., Cunningham, H, and Wilks, Y., MUlti-Source Entity recognition – an Information Extraction System for Diverse Text Types. Technical report CS--02--03, Univ. of Sheffield, Dep. of CS, 2003.
16. Mori, J.,, Matsuo, Y., and Ishizuka, M. Personal Keyword Extraction from the Web, Journal of Japanese Society of Artificial Intelligence, Vol.20, No.5, pp.337-345, 2005.
17. Resnik, P. Semantic Similarity in a taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. Journal of Artificial Intelligence Research
18. Sanderson, M., and Croft, W.B. Deriving concept hierarchies from text. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and Development in Information Retrieval, pages 206–213, 1999.
19. Sekine, S., Sudo, K., Nobata, Ch., Extended Named Entity Hierarchy (LREC 2002).
20. Turney, P.D. Coherent key phrase extraction via Web mining, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03).
21. Witten, I., Paynter, G., Frank, E., Gutwin, C. and NevillManning, C. KEA: Practical Automatic Key phrase Extraction. In the Proceedings of ACM Digital Libraries Conference, pp. 254-255, 1999.
22. Zhang, L., Yu, Y., Zhou, J., Lin, C.X. and Yang, Y. An Enhanced Model for Searching in Semantic Portals, in Proc. of 14th International World Wide Web Conference (WWW2005), May 10-14, 2005, in Chiba, Japan.
23. Zeng, H.J., He, Q.C., Chen, Z., Ma, W.Y., and Ma, J.W. Learning to Cluster Web Search Results. SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.

Automatic Annotation Using Citation Links and Co-citation Measure: Application to the Water Information System

Lylia Abrouk[1,2] and Abdelkader Gouaïch[1]

[1] LIRMM, Laboratoire d'Informatique,
de Robotique et de Microlectronique de Montpellier
161 rue Ada, 34392 Montpellier Cedex 5
{abrouk, gouaich}@lirmm.fr
http://www.lirmm.fr

[2] EMWIS, Euro-Mediterranean Information System
on the know-how in the Water sector
2229, route des cretes, 06560 Valbonne
l.abrouk@semide.org

Abstract. This paper describes an approach to automatically annotate documents for the Euro-Mediterranean Water Information System. This approach uses the citation links and co-citation measure in order to refine annotations extracted from an indexation method. An experiment of this approach with the CiteSeer database is presented and discussed.

1 Introduction

The Web offers technologies to share knowledge and information between organisations and users that can be distributed world-widely. In this paper we discuss the use of Web technologies for a specific professional domain that is sharing information on water management among Mediterranean countries that participate to the Euro Mediterranean Information System on the know how in the water sector (EMWIS, www.emwis.org).

EMWIS is an information and knowledge exchange system between the Euro Mediterranean partnership countries, necessary for the implementation of the Action Plan defined at the Euro Mediterranean Ministerial Conference on Local Water Management held in Turin in 1999. The objectives of EMWIS are as follows:

- Facilitate the access to information on water management;
- Develop the sharing of expertise and know-how between the partnership countries;
- Elaborate common outputs and cooperation programs on the know-how in the water field.

Using Web technologies within EMWIS to make information available is necessary but far from being sufficient. In fact, information is useful only when it can be retrieved later by users that need it. However, searching for the most

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 44–57, 2006.
© Springer-Verlag Berlin Heidelberg 2006

relevant information that meets user's request is still a problem especially when informations are coming from heterogeneous sources and sometimes accessible only with some rights. To solve this problem, informations, that are abstracted as *resources*, are annotated to describe both: *(i)* their context of creation: names of the authors, date of appearance and so on; *(ii)* and the semantics of their content.

The annotation of resources is very useful in order to match users' requests with resources that are available within EMWIS. However, annotating manually all the resources in a large system such as EMWIS is infeasible.

In this paper, we present an approach in order to annotate automatically a set of unannotated resources by using citation links. By contrast with classical Web approaches for automatic annotation, we use a restrained vocabulary of annotation defined in the EMWIS's global ontology.

The rest of the paper is organised as follows: Section 2 presents the context of our work and states the problem treated in this paper; Section 3 presents the backgrounds of works that have already used link analysis for different purposes such as statistical analysis, classification of resources, and meta-data propagation; Section 4 presents our approach in order to automatically annotate resources using citation links; Section 5 presents our experimentation with the Citeseer data base; and finally, Section 6 presents some perspectives and conclusions.

2 Context and Problematic

The global architecture of EMWIS defines the following entities: a National Focal Point (NFP) for each country and a single Technical Unit (TU). The NFPs are restrained teamworks that:

- create and make available a national server to access information;
- handle and manage the information system's national users.

The TU acts as a facilitator in helping each NFP to set up their information system and ensuring the coordination among all the NFPs. It is worth noting that the architecture of the EMWIS is fully distributed and Web technologies are used to share information among all EMWIS entities.

Figures 1 presents an example where a user searches some resources (documents in this case) on a specific theme. The documents are distributed among all the NFPs. To answer the user's request we face a first problem that is related to the description. In fact, the documents have to be well described by using all possible languages spoken within the EMWIS participating countries. To avoid this problem we have considered a common vocabulary to describe the resources. This is known as the EMIWIS global ontology. We have also to consider that this ontology is not static and can be updated by adding new concepts.

To implement technically EMWIS objectives, we have considered the following goals for our work:

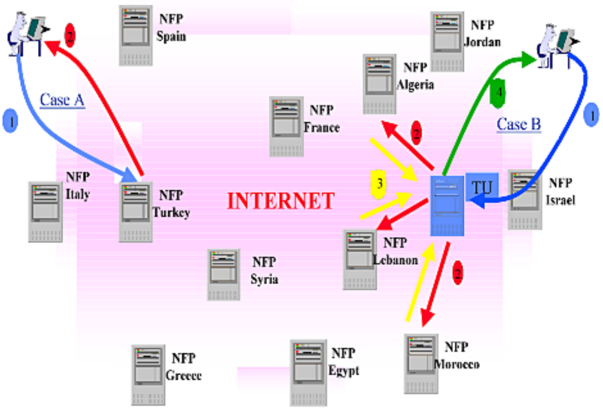


Fig. 1. EMWIS Architecture

1. resource annotation: in this part, we are focused on how to annotate automatically resources that have not been annotated by the experts of the domain;
2. global ontology enhancement: in this part, we are focused on how to add new concepts and relationships within the global ontology and how to update automatically the existing annotation of resources.

This paper targets only the first part of the work and presents means in order to annotate automatically a large set of resources using the citation links that structurally exist among resources. In fact, within a large system like EMWIS it is not feasible to assume that the content of all the resources can be described manually by experts. Our goal is then to provide a mean to assist experts and content managers to annotate the resources by suggesting automatically some annotations after analysing the citation links of already existing resources.

3 Backgrounds

Before presenting the state of the art, we provide some general definitions that will be used in the rest of the paper:

1. A document is the material that supports the encoding of information. The document can be either a hard copy, a web page, or any other medium that makes information persistent.
2. A resource is a generic concept that we use in order to talk about documents when these documents are needed to be used. There are several relationships between resources such as: citation, access link (for instance hyperlinks).
3. A citation occurs between documents: in this case, the document that *cites* another document, indicates that it is 'talking about' some parts of the *cited* document.

4. An access link, or hyperlink, indicates that the *target* document can be accessed directly from the *source* document.

This section presents works that have already used the relationships among the resources in order to:

- Extract statistical information;
- Classify the documents according to their importance;
- Propagate annotations and meta-data among documents.

3.1 Bibliometry

The Bibliometry is a statistical analysis of scientific publications [11]. It provides some qualitative and quantitative mesures about the activity of producers (scientists, laboratories and so on) and broadcasters (journals, editors and so on) of scientific documents.

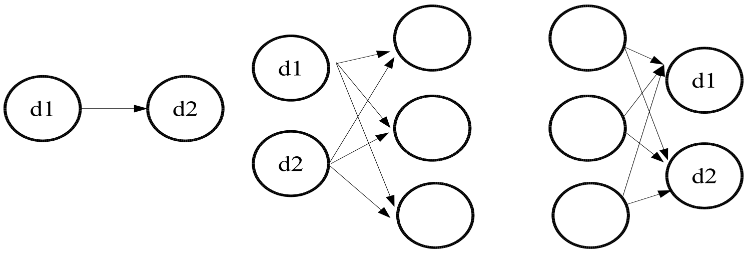
The bibliometry field considers the citations among the documents: *citation analysis*. The citation analysis is about establishing relations between the authors and documents and defining other more complex relations such as the co-reference and co-citation. These relationships are described in more details in the next paragraph.

3.2 Citation Analysis

Scientific documents can be modelled as an directed graph $G = (N, A)$ where the nodes represents articles and the arrows citation relationship.

Figure 2 illustrates some relationships among documents:

- Citation relationship: when a document d_1 references a document d_2 for instance. Generally the citation analysis determines the impact of one author on a given field by determining the amount of time that this author is cited by others.



(A) d1 cites d2

(B) Bibliographic coupling of d1 and d2

(C) co-citation of d1 and d2

Fig. 2. Relations among documents

- [8] has introduced the *bibliographic coupling* relationship. Documents are considered as bibliographically coupled when they share one or more bibliographic references. However, the bibliographic coupling is now displaced by co-citation clustering.
- The co-citation relationship represents documents that are cited by the same documents. The co-citation method [5], that has been used in bibliometry since 1973, aims to create relationships between the documents that are in the same domain field or *theme*. The hypothesis is that documents which are cited jointly share the same theme.

3.3 Propagating Meta-data Using Links

Marchiori [12,13] has used link analysis to propagate meta-data among documents (Web pages). His idea is that when a document d_1 owns some meta-data (or keywords) (a_v) (which indicates that the keyword a has a weighting equal to v) and there is a document d_2 with a hyperlink to d_1, then the keywords of d_1 are propagated to d_2 but with a loss factor, f, such that the keywords are equal to $(a_{v \times f})$. The same mechanism is then applied to all pages that are linked to d_2. This time, the resulting keywords weighting will be $(a_{v \times f \times f})$. Consequently, the keywords of the initial page d_1 are propagated to all accessible and indirectly accessible pages with a loss factor until reaching a defined threshold.

Prime [17] has also used links in order to propagate meta-data among documents. The core idea of this work is to add nonthematic meta-data to thematic meta-data that have been added by search engines. As Marchiori, Prime considers that when a link exists between two documents then these documents share the same thematic. However, Prime does not propagate meta-data using directly the Web graph but by using a subset called *co-citation graph.* The first step of this methodology is to determine the similarity between Web documents using a similarity index: two pages are close according to their citation frequency and co-citation frequency. The second step gathers closest pages in clusters.

3.4 Link Analysis for the Web

The classification of web pages is a known example that uses link analysis to find most important pages. The most known algorithms are: *Page Rank* [2,3,14] and *HITS*[9,7].

The *Page Rank* algorithm is used by Google[1] to classify web pages. The principle of this approach is to consider that a page is more important if there are several pages that point on it. This measure assumes three hypothesis [1]:

1. the popularity of a page depends on the popularity of the pages that point on it;
2. the links of a page do not have the same importance;
3. the popularity of a page does not depend on the users' requests.

[1] www.google.com

The *Hits* algorithm uses a search engine to identify in a set of web pages *authorities* and *hubs*. Hubs and authorities exhibit what could be called a *mutually reinforcing relationship*. A good hub is a page that points to many good authorities; a good authority is a page that is pointed to by many good hubs. An iterative algorithm HITS (Hypertext Induced Topic search) calculates the values of hubs and authorities for each page. The first step consists of posting a query; Hits assembles a root set s of pages, returned by the search engine on that query; it expends then this set to a larger base set t by adding in any pages that point to, or are pointed by, any page on s. The second step is to associate with each page a *hub-weight* and an *authority-weight*. The update operations are performed for all the pages, and the process is repeat until convergence that is proven to be reached.

We can also mention other uses of links such as for: *(i)* geographical categorisation of Web pages such as in work of [4]. *(ii)* discovery of similar Web documents, for instance [15] calculates using links the level of similarity between Web documents; *(iii)* discovery of communities in the Web such as in work of [7,10,19].

4 Automatic Annotation of EMWIS Documents

Before describing our approach, the types and the organisation of EMWIS documents are presented. EMWIS documents can be one of the following types: news, event document, legal documents, technical document, slide presentation document, Web document. The events are seminars, workshops, conference, courses that are organised by EMWIS.

For an event there is a Web document that includes a description and links to other documents related to this event. In the rest of the paper, when there is no ambiguity the term 'event' is used directly to talk about the 'event document'. Each event cites other documents, such as the Web page of the NFP that organises the event, a document that describes the topic of the event, and a set of presentations and publications. Most of the EMWIS documents are not annotated and this task is impossible to perform manually.

Section 2 has presented two main questions related to: *(i)* the uniform description of documents to avoid translating annotations in each language; *(ii)* the annotation of all documents using terms defined in the global ontology.

To answer the first question, we have defined a global ontology of the EMWIS community. This ontology is a set of concepts structured as a tree. The links among the concepts are semantic relationships (synonymy, aggregation, composition) or inheritance. To each concept we associate a set of terms in each language. Figure 3 describes a small part of the EMWIS ontology.

Figure 4 describes the major steps for the annotation and the global ontology enhancement processes:

1. For the document d a first annotation is generated using an indexation method. The result is a set of concepts belonging or not to the global ontology. Let E_{og} be the set of concepts that belong to the global ontology and

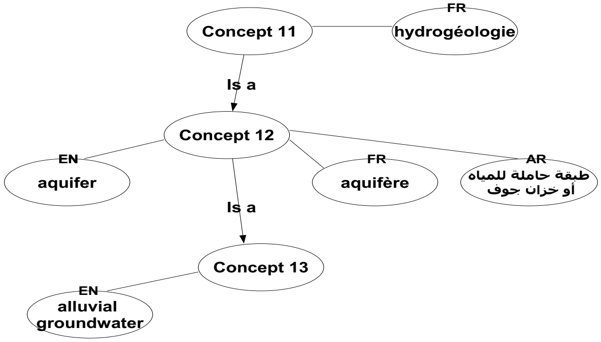


Fig. 3. EMWIS ontology

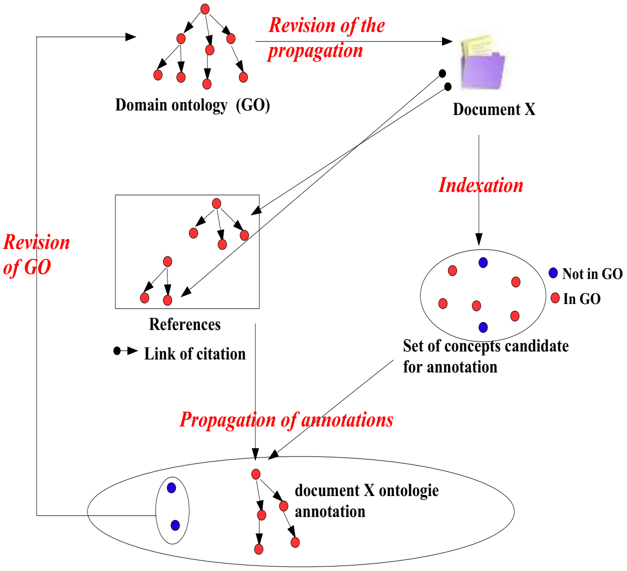

Revision of the propagation
Domain ontology (GO)
Document X
Indexation
Revision of GO
References
Link of citation
Not in GO
In GO
Set of concepts candidate for annotation
Propagation of annotations
document X ontologie annotation

Fig. 4. Global solution

E_{ong} the set of concepts that do not belong to the global ontology. The result of the annotation of a document is then $E_{og} \bigcup E_{ong}$. It is worth noting that the annotation generated by the indexation is not precise enough to describe to content since it contains a lot of terms and noises.

2. On the basis of the assumption that all the documents are annotated by using only concepts of the global ontology, the second step refines the first annotations by using the annotations coming from the citations of d. This is performed by adding or removing concepts from the set E_{og}. This step is known as *the propagation of the annotations*.

3. The third step which is the enhancement of the global ontology updates the global ontology by concepts defined in E_{ong}.
4. The update of the global ontology might imply the revision of the propagation process (step 2).

This article is focused only on the propagation of the annotations. So, having the structure of EMWIS documents, we suggest to use the citation links ,similarly too [13] and [17], to select meaningful annotations. To implement this solution, one has to answer the following questions: *(i)* what citations should be taken into account? In fact, not all the citations in a document are meaningful to determine the theme of the document; *(ii)* How to annotate the document? *(iii)* and finally, how to merge annotations that come from the selected documents.

The answer of these questions is provided by the following steps:

- structuring the documents using the co-citation analysis;
- selecting a subset of cited documents;
- importing and selecting the annotations which are coming from the selected documents.

4.1 Building the Co-citation Graph

When an author cites another document, this is done to indicate that the cited document contains some information that relevant to the context of the citation. However, we can also find citations that contribute to a small part of the document and do not necessarily determine the general theme of the whole document. Consequently, we have to consider only citations that contribute to determine the thematic of the source document. The co-citation method has been proven to be a good measure to determine the similarity on theme among documents. In fact, when documents are often cited together by different documents, we can assume that they target the same subject. We use the similarity index as described by [16] as follows:

$$SI_{(i,j)} = \frac{C^2_{(i,j)}}{C_{(i)} * C_{(j)}}$$

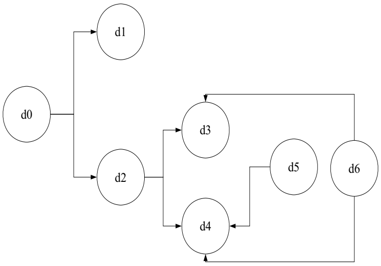


Fig. 5. An example of a citation graph between documents

- $C_{(i,j)}$ is the co-citation frequency or the number of time that i and j are cited together;
- $C_{(i)}$ is the citation frequency or the number of time that the page i is cited;
- $C_{(j)}$ is the citation frequency or the number of time that the page j is cited.

A distance function $d(i,j)$ is then defined as $d(i,j) = 1SI_{(i,j)}$. Using this distance function the co-citation matrix and graph are built as shown by the example presented in Figure 5.

The co-citation matrix of the example presented in Figure 5 is:

$$\begin{pmatrix} 0.00 & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 \\ 1.00 & 0.00 & 0.50 & 1.00 & 1.00 & 1.00 & 1.00 \\ 1.00 & 0.50 & 0.00 & 1.00 & 0.83 & 1.00 & 1.00 \\ 1.00 & 1.00 & 1.00 & 0.00 & 0.33 & 1.00 & 1.00 \\ 1.00 & 1.00 & 0.83 & 0.33 & 0.00 & 1.00 & 1.00 \\ 1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 0.00 & 1.00 \\ 1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 0.00 \end{pmatrix}$$

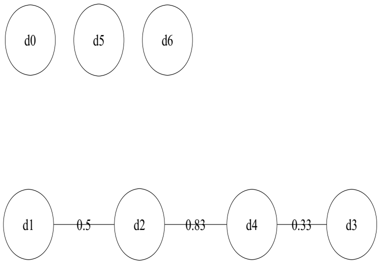


Fig. 6. The co-citation graph of the example presented in Figure 5

This matrix defines the co-citation graph that is presented by Figure 6: documents are linked with a weighted link that is equal to the co-citation distance; values equal to 1 are ignored.

The next step is to determine some clusters by defining a threshold distance f. The maximum distance following the paths within a cluster cannot exceed this threshold. We use classical clustering methods in order to have clusters with maximum documents and were the maximum distance between the documents following paths do not exceed the threshold. For instance, when $f = 0.5$ then we build two clusters as presented in Figure 7. These clusters are interpreted as themes were the documents are aggregated on.

4.2 Selecting the Meaningful Citations

Figure 8 presents a case when a new document is being included to the system. d_7 cites some exiting documents : $\{d1, d3, d4\}$. We assume then a document can

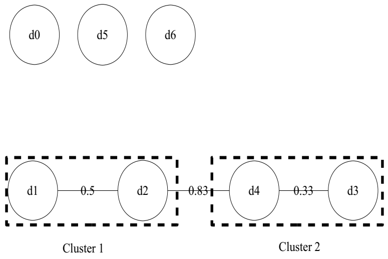


Fig. 7. Clusters with a threshold set to $f = 0.5$

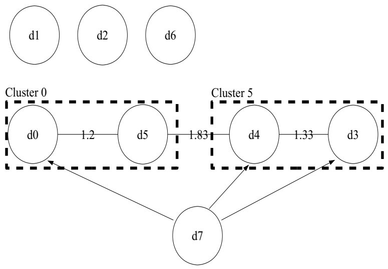


Fig. 8. Adding a new document d_7 to the system

target more that one theme. So, one has to provide a mean in order to select which citations are considered for the import. We suggest to define an order relationship between the clusters relatively to a document d:

$$cl_1 \leq_d cl_2 \equiv (\#\{cl_1 \cap citations(d)\}, \#cl_1) \leq (\#\{cl_2 \cap citations(d)\}, \#cl_2)$$

In this order relationship the first criterion considers the numbers of citations that belong to the cluster; the second order criterion considers the importance of the cluster.

When considering the previous example we have:

$$cluster_1 \leq_{d_7} cluster_2 \; as: \quad (1, 2) \leq (2, 2)$$

By using this order relationship an ordered list of clusters for the incoming document is created. We add another parameter that is the maximum number of themes allowed for a document: *max_theme*. The document that are selected for the annotation import are those that are cited by the article and that belong to the highest *max_theme*-clusters of the ordered list.

For instance, if we consider that $max_theme = 1$ for the simple example; then we select only documents that belong to $cluster_2$ and that are cited by c_7, which means $\{d4, d3\}$.

4.3 Selecting and Importing the Annotations

The last step has produced a set of articles for the import. However, one has to make a choice on: *(i)* what annotation to select; *(ii)* and what to import in an annotation knowing that every annotation is a tree of concepts defined within the EMWIS global ontology.

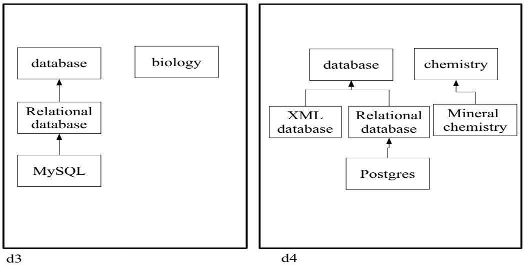


Fig. 9. An example of the annotation of documents d_3 and d_4

Figure 9 presents an example of annotation of documents d_3 and d_4. The naive solution would be to import the whole annotations of the document d_7. However, some annotation concepts are either not relevant with the content of d_7 or too specific for d_7. To solve these problems we use the result of the indexation process of the document. In fact, the indexation will generate the set of terms that appear frequently in the document. Consequently, only the intersection between in set of terms produced by the indexation and the concepts of the annotations is considered. This allows to remove concepts that are not found in the document and to select the right level in the annotation tree. For instance, if the term *'relational database'* appears several times within the document d_7, it will be produced by the indexation process. The intersection of this term with the annotations of d_3 and d_4 will remove the *'chimestry'* and 'biology' concepts as d_7 does not use these terms. Concepts that are too specific to d_3 and d_4 such as the type of database system used (MySQL, Postgres) will also be remove since the intersection stops the depth of the tree to *'relational database'*. So, the final annotation of d_7 will be the tree starting by the concept *'database'* and until the concept *'relational database'*. It is worth noting that in this simple description of the example we have simplified the process. In fact, we have used some means (such as synonyms and so on) in order to map indexation results , which are general terms, to the concepts of the global ontology.

5 Realisations and Experiments

To experiment with our approach we have considered the CiteSeer[2] collection as a test database. CiteSeer [18],[6] is a digital library for scientific literature. CiteSeer localises scientific publications on the Web and extracts some information

[2] http://citeseer.ist.psu.edu/

such as citations, title, authors and so on. This collection has been selected for two reasons: *(i)* the important number of documents; *(ii)* the fact that is contains scientific documents that use several citations. We have built a database that contains more than 550 000 documents.

However, CiteSeer description of documents cannot be used directly. In fact, CiteSeer uses a general vocabulary to describe the content of a document. But, we are interested only to description of documents using a controlled vocabulary or an ontology. We have used the ACM controlled vocabulary as an ontology to annotate CiteSeer documents during the experiment.

The presented approach has been implemented and the automatic annotation of unannotated articles has been performed. The experimentations show that the indexation keywords have been considerably refined when considering the citations of the document. Furthermore, for a concept, x,that has been selected for the annotation the fact that all its parents will be included for the annotation this adds information that can be useful during the search. In fact, if the user request is not directly related to the concept x but about his father concepts, then the document will be selected as potentially interesting for the user.

For the CiteSeer database we are remarked that the co-citation graph naturally express the clusters and themes so there is not need for the f parameter. In fact, this parameter has been expressed by [16] to split clusters on themes but of a specific domain all the documents are transitively cited together express a cluster for a specific theme. We have also remarked that setting *max_theme* higher to 3 does not affect the results on annotations. This can be explained by the fact that scientific and technical papers targets specific themes and do not uses more than 3 themes.

However, we have not defined until now an objective evaluation method to prove the efficiency of our approach. In fact, all the evaluations are subjectives and tries to compare the automatic annotation with the annotations of a human expert. As a perspective, we have to provide an objective method as an evaluation of this approach. This problem can be faced in almost all similar works on the same field.

6 Conclusion

This paper has described an approach in order to automatically annotate documents used by a specific community, namely EMWIS users. Annotating manually all documents in a distributed and large information system is a hard task and the classical indexation methods generate too fuzzy and imprecise keywords. We have exploited the citation relationships an information about the context of the document in order to refine its annotation and to add general the concepts defined within the ontology of the domain.

The work that has been presented is this paper differs from other works that target general and open communities such as the Web. In fact, we address here a specific quite close community, which facilitate the elaboration of an ontology of the domain. This makes the annotations independent from the multiple languages spoken within the community and helps also for the searching the appropriate documents that meet users' requests by structuring the search from the specific to the general concepts of the annotations.

The experiments with the CiteSeer database has shown the feasibility of the approach and have allowed the automatic annotation of scientific articles. However, we still need an objective measure to evaluate our approach independently from human experts.

References

1. F. Aguiar. *Modélisation d'un systme de recherche d'information pour les systmes hypertextes. Application la recherche d'information sur le World Wide Web.* PhD thesis, Ecole suprieure des Mines de Saint-Etienne, 2002.
2. A. Arasu, J. Novak, A. Tomkins, and J. Tomlin. Pagerank computation and the structure of the web: Experiments and algorithms, 2001.
3. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, pages 107–117, Australia, 1998.
4. O. Buyukkokten, J. Cho, H. Garcia-Molina, L. Gravano, and N. SHivakumar. Exploiting geographical location information of web pages. In *WebDB (Informal Proceedings)*, 1999.
5. E. Garfield. Co-citation analysis of the scientific literature: Henry small on mapping the collective mind of science. *Essays of an Information Scientist: Of Nobel Class, Women in Science, Citation Classics and Other Essays*, 15(19), 1993.
6. S. Ghita, N. Henze, and W. Nejdl. Task specific semantic views: Extracting and integrating contextual metadata from the web. In *In Submitted for publication, L3S Technical Report*, 2005.
7. D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *UK Conference on Hypertext*, 1998.
8. M. Kessler. Bibliographic coupling between scientific papers. In *American Documentation*, pages 10–25, 1963.
9. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Journal of the ACM*, pages 139–146, 1999.
10. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *Computer Networks*, Amsterdam, Netherlands, 1999.
11. P. Lauri. The bibliometrics a trend indicator. In *International Journal Information Sciences for Decision Making*, page 2836, 1997.
12. M. Marchiori. The quest for correct information of the web: hyper search engines. In *The Sixth International WWW Conference*, Santa Clara, USA, April 1997.
13. M. Marchiori. The limits of web metadata, and beyond. In *Proceedings of the Seventh International World Wide Web Conference*, pages 1–9, Australia, 1998.
14. R. Mihalcea, P. Tarau, and E. Figa. Pagerank on semantic networks, with application toword sense disambiguation. In *Proceedings of the 20th international conference on computational linguistics (COLING2004)*, Geneva, Switzerland, 2004.
15. D. Phelan and N. Kushmerick. A descendant-based link analysis algorithm for web search. 2002.

16. C. Prime-Claverie. *Vers une prise en compte de plusieurs aspects des besoins d'information dans les modèles de la recherche documentaire : Propagation de métadonnées sur le World Wide Web.* PhD thesis, Ecole suprieure des Mines de Saint-Etienne, 2004.
17. C. Prime-Claverie, M. Beigbeder, and T. Lafouge. Propagation de métadonnes par l'analyse des liens. In *Journés Francophones de la Toile - JFT2003*, France, juillet 2003.
18. J. Stribling, I. G. Councill, J. Li, M. F. Kaashoek, D. R. Karger, R. Morris, and S. Shenker. Overcite: A cooperative digital research library. In *International Workshop on Peer-to-Peer Systems*, 2005.
19. V. Vandaele, P. Francq, and A. Delchambre. Analyse d'hyperliens en vue d'une meilleure description des profils. In *Proceedings of JADT 2004, 7es Journes internationales d'Analyse statistique de Donnes Textuelles*, 2004.

Semantic Annotation Using Horizontal and Vertical Contexts

Mingcai Hong, Jie Tang, and Juanzi Li

Department of Computer Science & Technology, Tsinghua Univ., Beijing, 100084. China
{hmc, tj, ljz}@keg.cs.tsinghua.edu.cn

Abstract. This paper addresses the issue of semantic annotation using horizontal and vertical contexts. Semantic annotation is a task of annotating web pages with ontological information. As information on a web page is usually two-dimensionally laid out, previous semantic annotation methods that view a web page as an ’object’ sequence have limitations. In this paper, to better incorporate the two-dimensional contexts, semantic annotation is formalized as a problem of block detection and text annotation. Block detection is aimed at detecting the text block by making use of context in one dimension and text annotation is aimed at detecting the ‘targeted instance’ in the identified blocks using the other dimensional context. A two-stage method for semantic annotation using machine learning has been proposed. Experimental results indicate that the proposed method can significantly outperform the baseline method as well as the sequence-based method for semantic annotation.

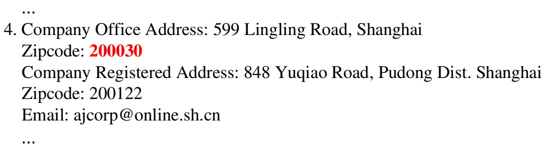
1 Introduction

Semantic web requires annotating existing web content according to particular ontologies, which define the meaning of the words or concepts in the content [1]. In recent years, semantic annotation has received much attention in the research community. Many methods have been proposed, for example, manual annotation, rule learning based annotation, and machine learning based annotation.

Conversional automatic annotation methods typically convert the web page into an ‘object’ sequence and utilize information extraction (IE) techniques to identify a subsequence that we want to annotate (i.e. targeted instance). (Here, the object can be either natural language units like token and text line, or structured units indicated by HTML tags like “<table>” and “<image>”). However, information on a web page is usually two-dimensionally laid-out and should not be simply described as a sequence. Figure 1 shows an example of document.

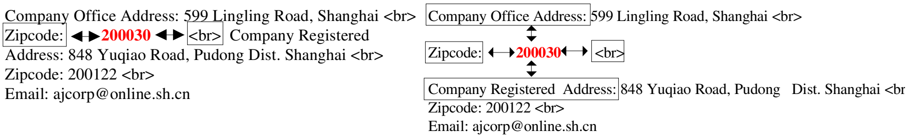
In this example, the targeted instance is the highlighted text “200030”. In terms of the sequence-based method, the snippet can be viewed as a token sequence and the task is to identify the sub token sequence “200030” (cf. Figure 2 (a), where “<br>” indicates a line break). In the identification, a usual approach will identify the start position and the end positions based on the context prior to and next to the targeted instance, e.g. “Zipcode:” and “<br>”. Unfortunately, in the example, the method will confuse the text “200122” with “200030” because they have the same context.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 58 – 64, 2006.
© Springer-Verlag Berlin Heidelberg 2006



...
4. Company Office Address: 599 Lingling Road, Shanghai
Zipcode: **200030**
Company Registered Address: 848 Yuqiao Road, Pudong Dist. Shanghai
Zipcode: 200122
Email: ajcorp@online.sh.cn
...

Fig. 1. Example of document


Company Office Address: 599 Lingling Road, Shanghai <br>
Zipcode: 200030 <br> Company Registered
Address: 848 Yuqiao Road, Pudong Dist. Shanghai <br>
Zipcode: 200122 <br>
Email: ajcorp@online.sh.cn
Company Office Address: 599 Lingling Road, Shanghai <br>
Zipcode: 200030 <br>
Company Registered Address: 848 Yuqiao Road, Pudong Dist. Shanghai <br
Zipcode: 200122 <br>
Email: ajcorp@online.sh.cn

(a) One-dimensional context (b) Two-dimensional context

Fig. 2. One-dimensional context vs. Two-dimensional context

An alternative method is to take into consideration of both the horizontal context and the vertical context (cf. Figure 2 (b)). For the targeted instance "200030", its vertical contexts (including above context "Company Office Address:" and below context "Company Registered Address:") can be used to distinguish it from instance "200012" and its horizontal contexts (including left context "Zipcode:" and right context "<br>") can be used to identify its start position and end position.

In this paper, to better incorporate the horizontal context and the vertical context, a two-stage method for semantic annotation is proposed in this paper. We formalize semantic annotation as that of block detection and text annotation. We propose to conduct semantic annotation in the two-stage fashion. We view the tasks as classification and propose a unified statistical learning approach to the tasks, based on Support Vector Machines (SVMs). The proposed method has been applied to a commercial project TIPSI, which is aimed at annotating the company annual reports from Stock Exchange. We used company annual reports from Shanghai Stock Exchange for experimentation. Experimental results indicate that the proposed two-stage methods perform significantly better than the baseline methods for semantic annotation. We observed +11.4% and +16.3% improvements (in terms of F1-measure) than the rule-based method and sequence-based method.

The rest of the paper is organized as follows. In section 2, we introduce related work. In section 3, we describe our approach to semantic annotation using horizontal and vertical contexts. In section 4, we use the annotation of company annual reports as a case study to explain one possible implementation. Section 5 gives our experimental results. We make concluding remarks in section 6.

2 Related Work

Related work can be summarized into three categories: annotation using rule induction, annotation as classification, and annotation as sequential labeling.

Many existing semantic annotation systems make use of rule induction to automate the annotation process (also called as 'Wrapper' induction, see [2]). For example, Ciravegna et al propose a rule learning algorithm, called LP^2, and have implemented an automatic annotation module: Amilcare [3]. The module can learn annotation rules from training data. The learned rules can then be used to annotate un-annotated documents. Amilcare has been used in several annotation systems, for instance, S-CREAM [4]. See also [5].

Another method views semantic annotation as classification, and automates the processing by employing statistical learning approaches (e.g. Support Vector Machines (SVMs) [6]). It defines features for each candidate instance and tries to learn a classifier that can detects the targeted instances from the candidate ones.

Different from the rule induction and the classification based methods, sequential labeling enables describing the dependencies between targeted instances in the semantic annotation. The dependencies can be utilized to improve the accuracy of the annotation. For instance, [7] proposes utilizing HMM in semantic annotation.

Much of the previous work converts the web page into an ‘object’ sequence (e.g. token sequence or text-line sequence) and utilizes information extraction (IE) techniques for identifying the targeted instance.

3 A Two-Stage Approach Using Horizontal and Vertical Contexts

In this paper, by *context*, we mean the surrounding information of the targeted instance. By *horizontal context*, we mean information left to and right to the targeted instance (e.g., the previous tokens and the next tokens). And by *vertical context*, we mean information above and below of the targeted instance (e.g., the previous lines and the next lines). For semantic annotation, we target at detecting the instances from a document and annotating each of the instances by a concept in a particular ontology.

We adopt a strategy of divided-and-conquer and formalize the problem of two-dimensional contexts based semantic annotation as that of block detection and text annotation. A *block* is a specific informative unit in a document. It can be defined by different granularity, e.g. text line, section, or paragraph. We also assign a label to each block. The assigned label corresponds to a concept in the ontology, implying that the block contain at least one instance of the concept. A block can have multiple labels indicating the block contains instances of different concepts. A block can also have no label (i.e. “*none*”) indicating that it contains no instance of any concept. The block can be laid horizontally or vertically. For facilitating the later explanation, we use vertically laid block as example hereafter.

In our two-stage approach, for block detection, a document is first viewed as a block sequence. For each block, we make use of its vertical context to detect its label. For text annotation, we view each identified block as an ‘object’ sequence and employ the horizontal context to detect the targeted instance.

In this work, we try to propose a general approach for semantic annotation. As case study, we work on annotating company annual reports. We only handle the annual reports in plain text format, i.e. non-structured data. We define a block as a text line, because in our experiments, statistic shows that 99.6% of the targeted instances are in one single text line (the statistic was conducted on the 3,726 experimental reports).

We formalize the two detection tasks as classification and employ a supervised machine learning approach. In block detection, we detect the label of each block using one classification model (the label corresponds to a concept in the ontology). In text annotation, we identify the start position and the end position of an instance using two classification models, respectively.

4 Annotating Company Annual Report Using Two-Stage Approach

To evaluate the effectiveness of the proposed approach, we applied it to a practical project TIPSI. In TIPSI, we are aimed at annotating the company annual reports from Shanghai Stock Exchange (SSE).

A company annual report generally consists of fourteen sections, including "Introduction to Company", "Company Financial Report", etc. A comprehensive annotation for the company annual reports should annotate company basic information, financial information, and directorate information, etc. Due to space limitation, we will only describe the annotation of the first part (i.e. Section "Introduction to Company") and omit details of the rests. Section "Introduction to Company" contains company information such as *Company-Chinese-Name*, *Legal-Representative*, *Company-Secretary*, and *Office-Address*. (See Section 5 for details.)

We make use of Support Vector Machines (SVM) as the classification model [6]. SVM-light, which is available at http://svmlight.joachims.org/, is employed in our experiments. We choose linear SVM in both block detection and text annotation tasks. We use the default values for the parameters in SVM-light.

In the rest of the section, we will explain processes of block detection and text annotation and feature definition in the two processes.

4.1 Block Detection

Detections of different types of blocks are similar problems. We view block detection as classification. For each concept, we train a SVM model to detect whether a block contains instance(s) of that concept. A text line is viewed as a block in this task. The key issue then is how to define features for effectively learning and detecting. In all detection models, we define features at token level and line level. We will take *ccn* as example to explain the feature definition in block detection models. Features used in *ccn* block detection model are:

Positive Word Features: The features represent whether or not the current line contains words like "公司' and "中文'. The words are usually used in the *ccn* block.

Negative Word Features: The features represent whether or not the current line contains words like "英文', "电 话'. These words are usually used in the other types of blocks and should not be included in the *ccn* block.

Special Pattern Features: A set of regular patterns is defined to recognize special patterns, such as email address, telephone number, fax number, URL. Each of the features respectively represents whether or not the current line contains one type of the special patterns.

Line Position Feature: The feature represents the line number of the current line. *ccn* block is usually placed in the first lines.

Number of Words Feature: The feature stands for the number of words in the current line.

The features above are also defined similarly for the previous line and the next line.

4.2 Text Annotation

An identified block contains at least one instance. We then try to identify the start position and the end position of the targeted instance. We view the problem as that of 'reverse information extraction' and employ two SVM models to perform the task. We also use the annotation of *ccn*'s instance as example in our explanation. Features used in *ccn* text annotation model are:

Token Features: The features respectively represent the specific tokens in the previous four positions, the current position, and in the next two positions. We define features using four previous tokens and only two next tokens. This is because our preliminary experiments show that the previous tokens seem more important in our annotation tasks.

Special Pattern Features: The features represent whether or not the current token contains a special pattern such as email address, telephone number, fax number, URL.

5 Experimental Results

5.1 Experiment Setup

We collected company annual reports from Shanghai Stock Exchange (http://www.sse.com.cn). We randomly chose in total 3,726 annual reports from 1999 to 2004. To evaluate the effectiveness of our approach, we extracted the Section "Introduction to Company" from each annual report for experiments.

In all the experiments, we conducted evaluations in terms of precision, recall and F1-measure. For block detection, we conduct evaluation at the line level. For the text annotation tasks, we perform evaluation at the 'instance' level.

We use the rule based annotation as baseline. The rules were defined according to the most useful features in the SVM models. For example, the rule to annotate *ccn* is "Token sequence starts after '*company Chinese name:*' and ends with '*Co., Ltd.*'".

We also compare the proposed approach with the sequence-based method. In this method, an annual report is viewed as a token sequence, and two SVM models are used to detect the start position and the end position, respectively. The same feature sets are used as that in the proposed approach for text annotation.

5.2 Experimental Results

We randomly split the data set into two 50:50 subsets, one for training and the other for testing. We then conducted the experiment in the following way. First, we used the SVM models to detect the type of each block (i.e. text line) and assign (a) label(s). Next, based on the output of block detection, we used two SVM models to detect and annotate the target instances. Block predicted as "*none*" were skipped. For each experiment, we repeated the split and conducted the experiments for ten times. We used the average results as the experimental result. We also made comparisons with the baseline methods described above.

Table 1 shows the experimental results on the data set. Baseline and Sequence denote the baseline method and the sequence-based method defined above, respectively. Our Approach denotes the proposed approach. Pre., Rec., and F1 respectively represent the precision, recall, and F1-measure.

Table 1. Performance of annual reports annotation (%)

Annotation Task		Pre.	Rec.	F1	Annotation Task		Pre.	Rec.	F1
Company Chinese Name (*ccn*)	Baseline	97.4	86.8	91.8	Registered Address (*caddr*)	Baseline	91.6	83.3	87.3
	Sequence	97.6	87.4	92.2		Sequence	86.3	63.9	73.6
	Our Approach	97.4	90.1	93.6		Our Approach	88.3	92.0	90.1
Company English Name (*cen*)	Baseline	74.1	70.1	72.0	Office Address (*coffice*)	Baseline	88.6	88.7	88.6
	Sequence	92.5	87.8	90.1		Sequence	83.6	64.0	72.5
	Our Approach	94.8	91.1	92.9		Our Approach	89.2	90.2	89.7
English Name Abbreviation (*ceabbr*)	Baseline	95.4	78.8	86.3	Zip of Office Address (*czip*)	Baseline	88.6	78.9	83.5
	Sequence	97.9	85.9	91.5		Sequence	73.7	93.9	82.5
	Our Approach	92.7	90.7	91.7		Our Approach	96.7	93.4	95.0
Legal Representative (*delegate*)	Baseline	93.4	92.2	92.8	Website (*curl*)	Baseline	91.2	69.1	78.6
	Sequence	96.0	94.7	95.4		Sequence	61.7	89.1	72.9
	Our Approach	95.8	96.8	96.3		Our Approach	90.3	93.0	91.7
Company Secretary (*sperson*)	Baseline	89.3	88.9	89.1	Email of Company (*cemail*)	Baseline	94.1	45.8	61.6
	Sequence	94.9	88.4	91.5		Sequence	89.6	34.7	50.1
	Our Approach	87.9	94.0	90.8		Our Approach	93.1	87.1	90.0
Tel. of Secretary (*stel*)	Baseline	88.8	75.4	81.6	Newspaper (*newspaper*)	Baseline	88.5	70.4	78.4
	Sequence	51.1	82.7	63.2		Sequence	97.8	95.2	96.5
	Our Approach	91.5	96.1	93.7		Our Approach	97.6	98.1	97.8
Fax (*sfax*)	Baseline	92.3	91.2	91.7	Stock Name (*sname*)	Baseline	88.3	77.0	82.3
	Sequence	55.5	83.9	66.8		Sequence	94.8	86.1	90.2
	Our Approach	96.3	96.5	96.4		Our Approach	91.2	95.3	93.1
Address of Secretary (*saddr*)	Baseline	92.2	91.3	91.7	Stock Code (*sno*)	Baseline	96.2	86.3	91.0
	Sequence	58.4	73.6	65.1		Sequence	94.5	90.3	92.3
	Our pproach	95.8	97.0	96.4		Our Approach	95.5	95.2	95.3
Email of Secretary (*semail*)	Baseline	75.1	81.0	77.9	Average	Baseline	89.7	79.7	83.9
	Sequence	41.4	66.2	50.9		Sequence	80.4	80.5	80.4
	Our Approach	93.8	95.2	94.4		Our Approach	**93.4**	**93.6**	**93.5**

We see that our method can achieve good performances in all the tasks. For each annotation task, our approach significantly outperforms the baselines as well as the sequence-based methods. Now, we make discussion for the experimental results.

(1) Improvements over baseline method. The baseline method suffers from low recall in most of the annotation tasks, e.g. *cemail*, *curl*, and *newspaper*, although its precision is high. This is due to a low coverage of the rules. Our approach outperforms the baseline method by 11.4% in terms of F1-measure. This also indicates that the features used in block detection and text annotation are effective.

(2) Two-dimensional context vs. One-dimensional context. In annotation of *ccn*, *cen*, *ceabbr*, *delegate*, *sperson*, *newspaper*, *sname*, and *sno*, the sequence-based method achieved high performance. This is because these fields are distinguishable by using only the horizontal context. While in the other annotation tasks, the sequence-based method suffers from lack of context and results in poor performance, even poorer than the baseline. It confirms us that accurate semantic annotation on company annual reports requires not only horizontal context, but also vertical context. Our approach benefits from the usage of both horizontal and vertical contexts.

(3) Error analysis. We conducted error analysis on the results of our approach.

In block detection stage, there are mainly three types of errors. The first type of errors was due to extra line breaks in the text, which mistakenly breaks the targeted instance into multiple lines. The second type of errors was because of extra spaces in the Chinese text (note space in the Chinese text space is different from that in the English text), e.g. "上海市零陵路' is mistakenly written as "上海 市零陵路".

In text annotation stage, errors can be summarized into two categories. The first type of errors was due to the errors at the block detection step. The second type of errors was due to errors of detection of instances' end position.

6 Conclusion

In this paper, we have investigated the problem of semantic annotation using horizontal and vertical context. We propose a two-stage approach on the basis of machine learning methods. The proposed approach has been applied to annotate company annual reports. Experimental results show that our approach can significantly outperform the baseline methods as well as the sequence-base methods.

References

[1] R. Benjamins and J. Contreras. Six Challenges for the Semantic Web. Intelligent Software Components. Intelligent software for the networked economy (isoco). April, 2002

[2] N. Kushmerick, D.S. Weld, and R.B. Doorenbos. Wrapper Induction for Information Extraction. In Proc. of IJCAI. Nagoya, Japan. 1997:729-737

[3] F. Ciravegna. (LP)2, an Adaptive Algorithm for Information Extraction from Web-related Texts. In Proc. of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining, Seattle, USA. August 2001

[4] S. Handschuh, S. Staab, and F. Ciravegna. S-CREAM—Semi-automatic Creation of Metadata, In Proc. of EKAW 2002, Siguenza, Spain, 2002: 358-372

[5] J. Tang, J. Li, H. Lu, B. Liang, and K. Wang. iASA: Learning to Annotate the Semantic Web. Journal on Data Semantic. 2005, Vol(4): 110-145

[6] C. Cortes and V. Vapnik. Support-Vector Networks. Machine Learning, Vol(20), pp273-297. 1995

[7] L. Reeve. Integrating Hidden Markov Models into Semantic Web Annotation Platforms. Technique Report. 2004

Semantic Wiki as a Lightweight Knowledge Management System

Hendry Muljadi[1], Hideaki Takeda[1], Aman Shakya[2], Shoko Kawamoto[1],
Satoshi Kobayashi[1], Asao Fujiyama[1], and Koichi Ando[3]

[1] National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, Japan 69121
{hendry, takeda, skawamot, satoshi-k, afujiyam}@nii.ac.jp
[2] Asian Institute of Technology, P.O.Box 4, Klong Luang, Pathumthani 12120
aman.shakya@ait.ac.th
[3] Shibaura Institute of Technology, 3-7-5, Toyosu, Koto-ku, Tokyo, Japan
andou@sic.shibaura-it.ac.jp

Abstract. Since its birth in 1995, Wiki has become more and more popular. This paper presents a Semantic Wiki, a Wiki extended to include the ideas of Semantic Web. The proposed Semantic Wiki uses a simple Wiki syntax to write labeled links which represent RDF triples. By enabling the writing of labeled links, Semantic Wiki may provide an easy-to-use and flexible environment for an integrated management of content and metadata, so that Semantic Wiki may be used as a lightweight knowledge management system.

Keywords: Wiki, MediaWiki, Semantic Wiki, metadata, RDF.

1 Introduction

Since its birth in 1995, Wiki has become more and more popular. It is a simple publishing system that is easy to learn and quick to use. In Wiki, people can create or edit a Wiki page using a simple syntax to write content [1]. The popularity of Wikipedia[1], the online encyclopedia, has proven how Wiki is effective for the collaboration on the web. Wiki has been considered for the development of a knowledge management system [2],[3]. It has gained significant attention from industry as well [4].

Nowadays there are so many Wiki engines available. One of the famous Wiki software is MediaWiki[2], which is being used to run Wikipedia. MediaWiki has the category management function which allows a Wiki page under the namespace ("Category:") to be used as a metadata, and also allows user to create class and sub-class relation, as well as class and instance relation between Wiki pages. In other words, MediaWiki has the capability to manage: (1) contents, (2) metadata, and (3) the relations between contents and metadata. However, the metadata is not suitable for processing by applications.

Nevertheless, in a Wiki environment, it is easy to make an Resource Description Framework (RDF) resource, since a Wiki page always has a URL, e.g. http://

[1] en.wikipedia.org/wiki/Wikipedia
[2] www.mediawiki.org

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 65 – 71, 2006.
© Springer-Verlag Berlin Heidelberg 2006

hostname/wiki/pagename, and this URL can be used as an URI of an RDF resource. RDF is a language to express metadata that is suitable for processing by applications [5]. It consists of subject-predicate-object triples that state specific facts about resources or concepts, e.g. “[Homer]<HasChild>[Bart]”, where subject, predicate and object (if not a literal) are identified via URIs. Constructing RDF triples in a Wiki environment can be done by enabling the construction of labeled links [6]. The labeled link represents the RDF property that links the RDF subject with its object.

For the development of a lightweight knowledge management system, a Wiki is extended to enable the writing of labeled links. This paper presents MewKISS, a Wiki extended to enable the integrated management of content and metadata. Section 2 presents the extension of Wiki to enable the integrated management of content and metadata. Section 3 shows the implementation cases of MewKISS as a lightweight knowledge management system.

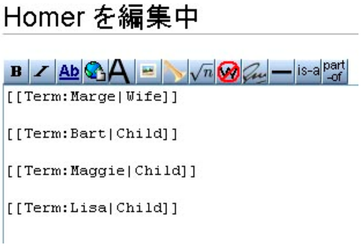
2 MewKISS

2.1 Semantic Extension of Wiki

MewKISS is an abbreviation for MediaWiki with Simple Semantics. It uses MediaWiki as its basis Wiki engine. By extending MediaWiki, the Semantic Wiki will have the benefit of having all the functions available in MediaWiki as a content management system.

Using the existing category management function as a reference, a new syntax is created to write the labeled links. Wiki syntax to write the labeled link is [[term:target_page|property]]. Each time this syntax is written on a Wiki page, the triple will be stored into the new table in the Wiki database.

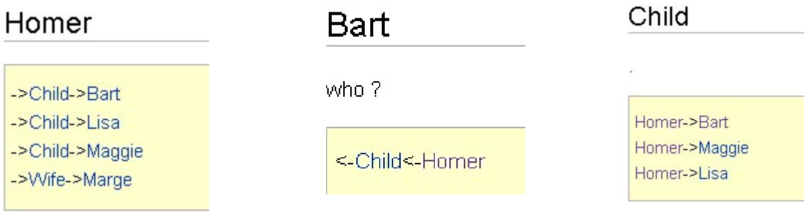
Fig.1 shows the example of the Wiki syntax writing on a Wiki page. The Wiki page on which the syntax is written will become the source page of the RDF triple. Fig.2(a), (b), (c) show how the labeled link relations are displayed on the source page, target page and property page respectively. Displaying labeled link relation allows users to navigate the relation between pages easily.


Homer を編集中

```
[[Term:Marge|Wife]]

[[Term:Bart|Child]]

[[Term:Maggie|Child]]

[[Term:Lisa|Child]]
```

Fig. 1. Using the Wiki syntax for RDF triple construction

Enabling MediaWiki to write labeled links with simple syntax allows users to create and manage relations between Wiki pages easily and flexibly. The writing of labeled links allows users to write and edit RDF triples even though they have no knowledge about it. Thus, MewKISS can be used as an integrated content and metadata management system.



(a) Display of the source_page ->property->target_page (b) Display of the target_page <-property<-source_page (c) Display of the property page source_page->target_page

Fig. 2. Display on the Wiki pages

2.2 Mapping to Other Semantic Web Application

Semantic Wiki has becoming more and more popular. There are more than 20 prototypes available[3]. Surveying the current trend, Semantic Wikis have gone into two poles. The first one emphasizes the need to build a Semantic Wiki as a Semantic Web application. It is useful for domain experts, but will leave non-technical users away from it. The other one emphasizes user-friendliness, especially for non-technical users. It leaves the more technical Semantic Web aspect to other applications.

MewKISS emphasizes the user-friendliness of the Wiki engine. It is developed to allow non-technical users to manage metadata easily, and leaves the more technical aspects to external applications.

Fig.3 shows the overall structure of MewKISS. The RDF triples are stored in a table in the MewKISS database. The stored RDF triples can be exported to RDF database such as Sesame[4]. Using Sesame, users can explore the exported RDF triples (see Fig.4), make queries etc. In other words, MewKISS can be used as a bridge between non-technical users and Semantic Web technology.

3 Implementation

The proposed Semantic Wiki is developed for the development of a Web-based Japanese Biodictionary. Currently, it is also used for the development of a manufacturing feature library, a feature-based manufacturing information management system.

[3] http://www.semwiki.org/

[4] http://openrdf.org/

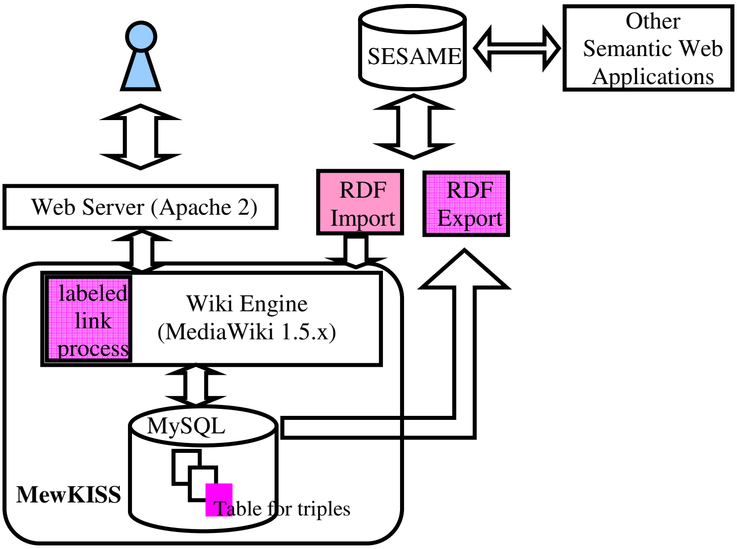
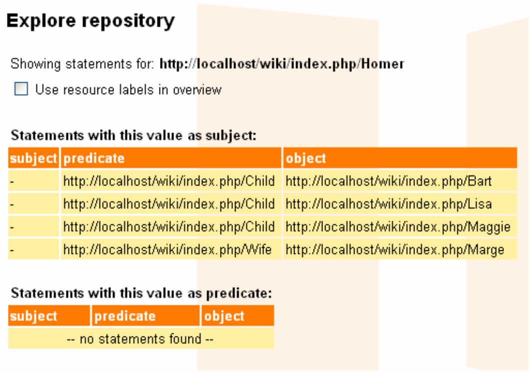


Fig. 3. The overall structure of MewKISS



Explore repository

Showing statements for: **http://localhost/wiki/index.php/Homer**

□ Use resource labels in overview

Statements with this value as subject:

subject	predicate	object
-	http://localhost/wiki/index.php/Child	http://localhost/wiki/index.php/Bart
-	http://localhost/wiki/index.php/Child	http://localhost/wiki/index.php/Lisa
-	http://localhost/wiki/index.php/Child	http://localhost/wiki/index.php/Maggie
-	http://localhost/wiki/index.php/Wife	http://localhost/wiki/index.php/Marge

Statements with this value as predicate:

subject	predicate	object
-- no statements found --		

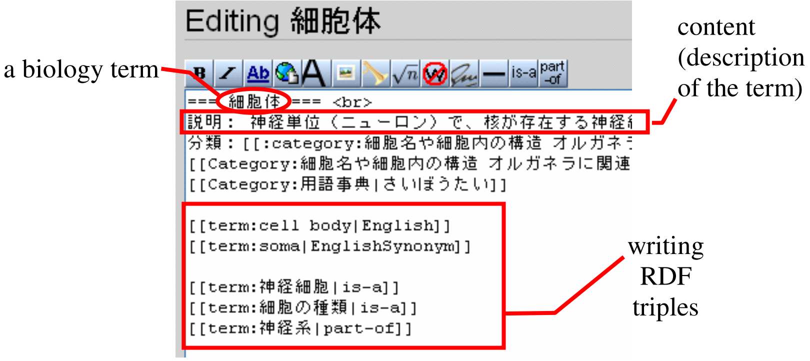
Fig. 4. Exploring the RDF repository

3.1 Semantic Wiki for the Development of a Web-Based Japanese Biodictionary

Developing a Web-based Japanese Biodictionary requires an environment where researchers from various biology fields may collaborate to create and maintain the content and the metadata of the dictionary. Semantic Wiki is able to provide such environment. It also provides the navigation support to manage relation between terms.

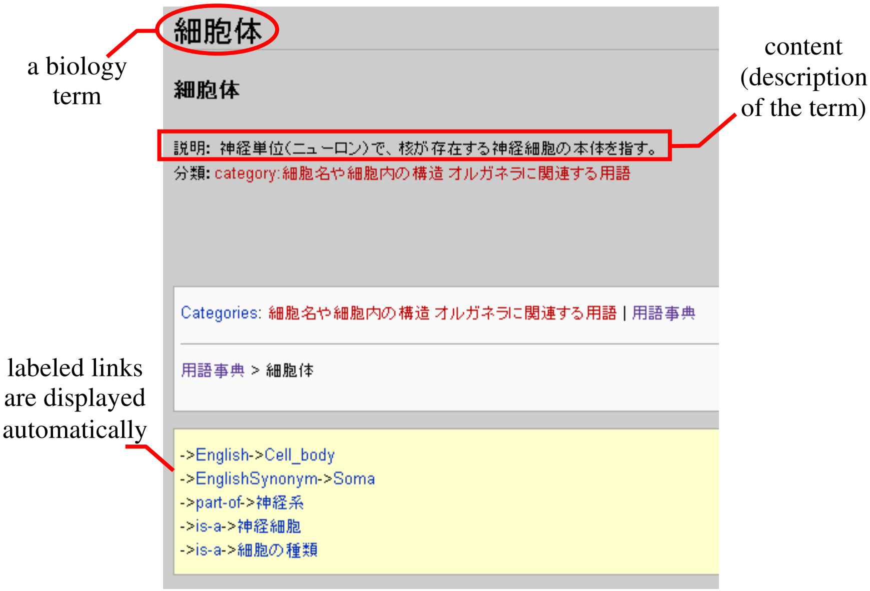
Currently, the prototype system contains more than 4,400 terms. Fig. 5 shows the editing page of a Japanese biology term. Users can write and edit the contents and the relation between pages easily and visually. In other words, users can write and edit

RDF triples even though they have no knowledge about it. Fig. 6 shows the Wiki page of the Japanese biology term. The labeled link relations that are written using the Wiki syntax are displayed. By displaying the labeled link relations, users can navigate the relation between terms visually.



```
=== 細胞体 === <br>
説明：　神経単位（ニューロン）で、核が存在する神経
分類：[[:category:細胞名や細胞内の構造 オルガネ
[[Category:細胞名や細胞内の構造 オルガネラに関連
[[Category:用語事典|さいぼうたい]]

[[term:cell body|English]]
[[term:soma|EnglishSynonym]]

[[term:神経細胞|is-a]]
[[term:細胞の種類|is-a]]
[[term:神経系|part-of]]
```

Fig. 5. Editing box of the Wiki page


a biology term
content (description of the term)
labeled links are displayed automatically

細胞体

細胞体

説明: 神経単位(ニューロン)で、核が存在する神経細胞の本体を指す。

分類: category:細胞名や細胞内の構造 オルガネラに関連する用語

Categories: 細胞名や細胞内の構造 オルガネラに関連する用語 | 用語事典

用語事典 > 細胞体

->English->Cell_body
->EnglishSynonym->Soma
->part-of->神経系
->is-a->神経細胞
->is-a->細胞の種類

Fig. 6. Wiki page of a biology term in the Japanese Biodictionary

3.2 Semantic Wiki for the Development of a Manufacturing Feature Library

In a feature-based process planning system, a manufacturing feature library plays an important role for the extraction of manufacturing information for the generation of process plans [7]. However, manufacturing technologies are progressing, and manufacturing information used in a particular factory may not be the same as the other factory. It is necessary to enable the management of the manufacturing information flexibly. In other words, it is essential to build a manufacturing feature library that is easy to modify and to customize. Semantic Wiki provides the solution.

For the development of the manufacturing feature library, a manufacturing feature ontology is created as the structure of the library (see Fig.7). This lightweight ontology is created as follows. First, manufacturing features such as step, slot etc are listed up. The manufacturing features used in this research are based on the library proposed by CAM-I [8]. Sub-classes of these manufacturing features are created by describing the manufacturing methods to create the shape of the parent classes. Sub-classes of these sub-classes are created by describing the tool types required for the manufacturing method. Instances of the lowest classes are created. Each instance contains specific manufacturing information, such as machine type, tool type, machining speed etc.

For the development of a Semantic Wiki-based manufacturing feature library, a new namespace ("MF:") is created. This namespace is used to handle manufacturing feature classes. Wiki syntax [[MF:feature_subclass|subclass]] is used to create class and sub-class relation of manufacturing features (see Fig.8). In the feature library, manufacturing information is also handled as Wiki pages. This allows the flexible management of manufacturing information, as well as the relation between the manufacturing feature's instance and the manufacturing information.

Semantic Wiki enables the creation of a structured manufacturing feature library. As data can be stored as RDF triples, the data can be processed by applications, which will support the automatic extraction of manufacturing information.

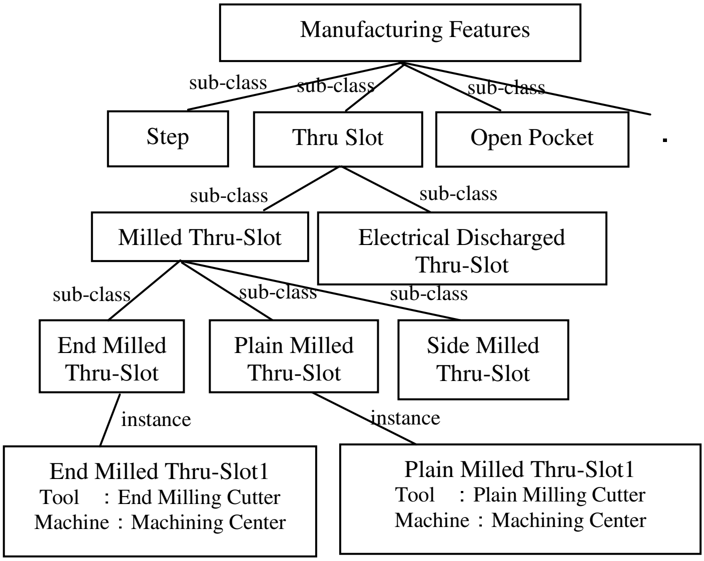
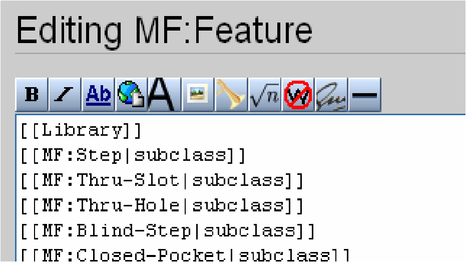


Fig. 7. A manufacturing feature ontology



Editing MF:Feature

```
[[Library]]
[[MF:Step|subclass]]
[[MF:Thru-Slot|subclass]]
[[MF:Thru-Hole|subclass]]
[[MF:Blind-Step|subclass]]
[[MF:Closed-Pocket|subclass]]
```

Fig. 8. Creating class and sub-class relation of the manufacturing features

4 Conclusion

The proposed Semantic Wiki is an extension of MediaWiki. It is able to write labeled links to construct RDF triples. It is a very simple software and as one tries to use this software, one may enjoy a visible editing of Wiki pages' relations. As it also inherits all the functions available in MediaWiki, it is a useful tool for the collaborative editing of contents and metadata according to simple RDF statements.

The implementation cases of the proposed Semantic Wiki show that it is an easy-to-use lightweight knowledge management system.

References

1. Leuf, B., Cunningham, W.: The Wiki Way: Quick Collaboration on the Web. Addison-Wesley, Boston (2001)
2. Wagner, C.: Wiki: a Technology for Conversational Knowledge Management and Group Collaboration. Communications of the Association for Information Systems, 13 (2004) 265-289
3. Raman, M., Ryan, T., Olfman, L.: Designing Knowledge Management Systems for Teaching and Learning with Wiki Technology, Journal of Information Systems Education, 16 (2005) 311-320
4. Cortese, A.: Business is Toying with a Web Tool. The New York Times, May 19 (2003)
5. Lassila, O.: Web Metadata: A Matter of Semantics, IEEE Internet Computing, Vol. 2, No. 4, (1998) 30-37
6. Takeda, H., Muljadi, H.: Towards Semantic MediaWiki. In Proc. of the 9th Semantic Web and Ontology SIG, Japanese Society for Artificial Intelligence. (2005) (in Japanese)
7. Ando, K., Muljadi, H., Takeda, H., Ogawa, M.: Development of Feature Library for a Process Planning System. In Proc. of the 8th Int'l Conf. on Manufacturing & Management. (2004) 885-890
8. Butterfield, W.R., Green, M.K., Scott, D.C., Stoker, W.J.: Part Features for Process Planning. Computer Aided Manufacturing International (CAM-I), Document R-86-PPP-01 (1988)

Partition-Based Block Matching of Large Class Hierarchies

Wei Hu, Yuanyuan Zhao, and Yuzhong Qu

School of Computer Science and Engineering, Southeast University,
Nanjing 210096, P.R. China
{whu, yyzhao, yzqu}@seu.edu.cn

Abstract. Ontology matching is a crucial task of enabling interoperation between Web applications using different but related ontologies. Due to the size and the monolithic nature, large-scale ontologies regarding real world domains cause a new challenge to current ontology matching techniques. In this paper, we propose a method for partition-based block matching that is practically applicable to large class hierarchies, which are one of the most common kinds of large-scale ontologies. Based on both structural affinities and linguistic similarities, two large class hierarchies are partitioned into small blocks respectively, and then blocks from different hierarchies are matched by combining the two kinds of relatedness found via predefined anchors as well as virtual documents between them. Preliminary experiments demonstrate that the partition-based block matching method performs well on our test cases derived from Web directory structures.

1 Introduction

Large-scale ontologies are a kind of ontologies created to describe complex real world domains. Large class hierarchies are one of the most common kinds of large-scale ontologies. Due to the decentralized nature of the Web, these large ontologies or class hierarchies for the same domain aren't unique. Examples can be found in: (a) Web directory structures, e.g., Google and Yahoo [1]; (b) product description standards, e.g., NAICS[1] and UNSPSC[2]; and (c) medicine or biology, e.g., GALEN[3] and FMA[4]. In order to achieve interoperation among Semantic Web applications using these large ontologies or class hierarchies, ontology matching is necessary. However, the size and the monolithic nature of these large ontologies or class hierarchies cause a new challenge to current ontology matching techniques. Therefore, some novel solutions are required.

In this paper, we propose a method for partition-based block matching that is practically applicable to large class hierarchies. Based on both structural affinities and linguistic similarities, two large class hierarchies are partitioned into

[1] http://www.naics.com

[2] http://www.unspsc.org

[3] http://www.opengalen.org

[4] http://sig.biostr.washington.edu/projects/fm

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 72–83, 2006.
© Springer-Verlag Berlin Heidelberg 2006

small blocks respectively, and then blocks from different hierarchies are matched by combining the two kinds of relatedness found via predefined anchors as well as virtual documents between them. Usually, structural affinities are computed by how closely they are related in the hierarchies, and linguistic similarities are computed by examining the similarities between the descriptions of the classes. The combinations of structural affinities and linguistic similarities are used to reflect the weighted links between the classes. Thus, large class hierarchies can be divided into small blocks base on an efficient linkage-based partitioning algorithm, e.g., ROCK [7]. Thereafter, two kinds of relatedness between blocks are found: one is via anchors which can be predefined by some simple methods or by experts; the other is via virtual documents [9]. These two kinds of relatedness are combined to match blocks in the end. The overview of the matching process is illustrated in Figure 1.

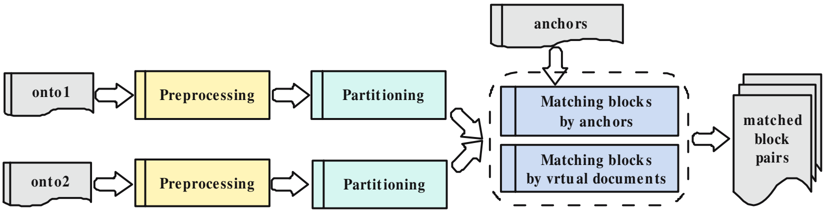


Fig. 1. The overview of the matching process

The rest of the paper is organized as follows: in the next section, some related works are introduced. In Section 3, we propose an efficient algorithm to partition large class hierarchies into small blocks. In Section 4, we present an approach to matching blocks. In Section 5, we show some preliminary experimental results to demonstrate the effectiveness of the method. In Section 6, we conclude with some directions for future work.

2 Related Work

Today, quite a lot of ontology matching or aligning approaches exist in literature, such as QOM [4], OLA [5], and V-Doc [9]. Please see [11] for a good survey about more representative approaches. However, most of these approaches have been developed for small-scale ontologies. For example, in V-Doc, when the ontologies to be matched have thousands of concepts, the matching process will take insufferably long time, and sometimes even cannot work. Another limit in current approaches is that most of them aim at 1:1 matching, not *block matching* (the relationship cardinality of the matching is many-to-many). Even in the field of schema matching, there are only a few works addressing the block matching issue, such as Artemis [2] and iMAP [3]. Artemis firstly computes 1:1 matching by using WordNet, and then generates block matching from the 1:1 matching by a hierarchical clustering algorithm. It is clear that this method is not targeted to large-scale ontologies because of its computational complexity for computing the 1:1 matching. iMAP semi-automatically discovers both 1:1 and complex mappings (e.g., room-price = room-rate $*$ (1 + tax-rate)). It exploits two new kinds of domain knowledge, i.e., overlap data and external data, to discover complex mappings. However, iMAP may be not a universal solution because it's not easy to specify the domain knowledge in some special cases.

The issue of partitioning large-scale ontologies (including large class hierarchies) has been recently addressed in [6,13,14], etc. In [6], an efficient solution for partitioning ontologies is provided by using ε-Connections. It guarantees that all concepts which have subsumption relations can be partitioned into one block, which becomes a limitation for ontology matching. In [13], large class hierarchies are automatically partitioned into small blocks. The background techniques are dependency graph and "island" algorithm. Although the main contribution of [14] is for ontology visualization, it also presents a method for ontology partitioning by Force Directed Placement algorithm. The main problem of these work is that they do not much concentrate on the sizes of blocks, so they do not well support ontology matching. For example, by applying ε-Connections to GALEN, we can gain only one block including nearly 10,000 concepts, which is not an appropriate size for ontology matching.

Compared with them, our method for partition-based block matching has three features: (a) it is efficient for large class hierarchies. In particular, the time complexity of the partitioning algorithm is $O(n^2)$; (b) it aims at block matching, because we believe it seems more useful for large class hierarchies than the current 1:1 matching; and (c) the sizes of most blocks are small enough to apply current ontology matching techniques to them.

3 Ontology Partitioning

In this section, we firstly introduce the notion of weighted links which are generated by combining two kinds of partitioning features extracted from large class hierarchies. Then, we present an efficient partitioning algorithm based on these weighted links.

3.1 Partitioning Features

In our investigation, large class hierarchies usually have two distinguishing characteristics: (a) they are often represented in DAG (Directed Acyclic Graph) structures and *is-a* relations are the most important built-in relations in large class hierarchies. An example is UNSPSC, it has 16500 classes, and the number of *rdfs:subClassOf* relations is 16500; and (b) linguistic similarities can be found between the local descriptions (e.g., local names, labels, comments) of the classes in these hierarchies. Therefore, two kinds of partitioning features can be extracted from large class hierarchies: one is structural affinities, which are based on (a); the other is linguistic similarities, which are based on (b).

Structural affinities between classes are defined by how closely they are related in the hierarchies, i.e., their structural closeness.

Definition 1 (Structural Affinities between Classes). *Let c_i, c_j be two classes. c_{ij} is the common superclass of c_i and c_j. $depthOf(c_k)$ returns the depth of class c_k in the class hierarchy. The structural affinity between c_i and c_j is defined as follows:*

$$aff_s(c_i, c_j) = \frac{2 \cdot depthOf(c_{ij})}{depthOf(c_i) + depthOf(c_j)}. \tag{1}$$

This equation has also been proposed in [14]. Please note that computing structural affinities between all the classes is time-consuming. Usually, only computing the affinities between the classes with adjacent depthes can obtain moderate results. In our experiments, we only compute structural affinities between the classes which satisfy $|depthOf(c_i) - depthOf(c_j)| \leq 1$.

Linguistic similarities are computed by examining the similarities between the local descriptions of the classes. Here, we adopt the string comparison method proposed in [12]. It considers that the similarity between two descriptions of two classes is related to their commonalities as well as to their differences.

Definition 2 (Linguistic Similarities between Classes). *Let d_i be the description of class c_i, d_j be the description of class c_j. The linguistic similarity between c_i and c_j is defined as follows:*

$$sim_l(c_i, c_j) = comm(d_i, d_j) - diff(d_i, d_j) + winkler(d_i, d_j), \tag{2}$$

where $comm(d_i, d_j)$ stands for the commonality between d_i and d_j, $diff(d_i, d_j)$ for the difference, and $winkler(d_i, d_j)$ for the improvement of the result using the method introduced by Winkler in [15].

The experimental results shown in [12] indicate that 0.65 is a good threshold, i.e., when $sim_l(c_1, c_2) < 0.65$, c_1 and c_2 would not be considered similar. In our experiments, we also find this threshold performs well in most scenarios, so we still take this threshold.

Finally, weighted links between classes are generated by combining the structural affinities and the linguistic similarities.

Definition 3 (Links between Classes). *Let c_i, c_j be two classes. ϵ_1 is a given threshold which satisfies $\epsilon_1 \in [0, 1)$. The weighted link between c_i and c_j is defined as follows:*

$$link(c_i, c_j) = \begin{cases} aff(c_i, c_j) & \text{if } aff(c_i, c_j) > \epsilon_1 \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

$$aff(c_i, c_j) = \alpha \cdot aff_s(c_i, c_j) + (1 - \alpha) \cdot sim_l(c_i, c_j), \tag{4}$$

where $\alpha \in [0, 1]$, and the selection of the parameter α depends on the structural and linguistic characteristics of the large class hierarchies.

We choose a small ϵ_1 for link filtering in our experiments, because the linkage among the classes is sparse, using a large ϵ_1 may cause many small “island” blocks, i.e., each block only contains several classes.

3.2 Partitioning Algorithm

Our partitioning algorithm is an agglomerative hierarchical partitioning algorithm mainly inspired by ROCK [7], which is a famous agglomerative clustering algorithm in the field of Data Mining. The main difference between ROCK and ours is that ROCK assumes that all the links between classes are the same; while we import the notion of weighted links, which reflect the information about the closeness between classes. Our algorithm accepts as input the set of n blocks to be clustered, which is denoted by B, and the desired number of blocks k, which is initially determined by application requirement. In each partitioning iteration, it selects the block having the maximum cohesiveness firstly, then choose the block having the maximum coupling with it, and finally merge these two blocks into a new block. The pseudo code of the algorithm is presented in Table 1.

Table 1. The partitioning algorithm

```
procedure($B, k$)
    for each block $B_i$ in $B$, do begin
        initialize the internal sum of links within $B_i$, called cohesiveness;
        initialize the sum of links between $B_i$ and others, called coupling;
    end
    while the number of current blocks $m > k$ do begin
        choose the best block $B_i$, which has the maximum cohesiveness;
        choose one block from the rest, which has the maximum coupling;
        merge block $B_i$ and $B_j$ named $B_p$;
        update $B_p$'s cohesiveness and coupling;
        remove $B_i$ and $B_j$;
        for each block other than $B_p$, update it's coupling;
        $m := m - 1$;
    end
end
```

The time complexity of this algorithm is $O(n^2)$. Compared with most other clustering or partitioning algorithms, it is quite efficient. Though k-means method is faster, it is worthy of noting that the means of the blocks are virtual entities, and if we change the means to the real entities (called k-medoids method), the time complexity also becomes $O(n^2)$ (e.g., PAM [8]). In addition, the centroid-based clustering algorithms aren't suitable for blocks of widely different sizes.

The most important point of the partitioning algorithm shown above is the computation of cohesiveness and coupling. Here, *goodness*() is used to compute the cohesiveness and coupling, and it measures the distance of two clusters by comparing the aggregate inter-connectivity of them.

Definition 4 (Goodness). *Let B_i, B_j be two blocks. $sizeOf(B_k)$ returns the number of the classes in B_k. The goodness between B_i and B_j is computed as follows:*

$$goodness(B_i, B_j) = \frac{\sum_{c_i \in B_i, c_j \in B_j} link(c_i, c_j)}{sizeOf(B_i) \cdot sizeOf(B_j)}, \quad (5)$$

when B_i, B_j are the same block, it computes the cohesiveness of the block; when B_i, B_j are two different blocks, it computes the coupling between them.

As pointed out in [7], choosing the denominator as $sizeOf(B_i) + sizeOf(B_j)$ is ill-considered. Though it may work well on well-separated blocks, in case of outliers or blocks with the classes that are neighbors, a large block may swallow other blocks and thus, the classes from different blocks may be merged into a single block. This is because a larger block typically would have a larger number of cross links with other blocks.

4 Block Matching

In this section, we present an approach to matching blocks. As shown in Figure 1, after partitioning pairwise large class hierarchies into two sets of small blocks respectively, we can find two kinds of relatedness between blocks from different sets: one is via predefined anchors; the other is via virtual documents [9]. The two kinds of relatedness are combined to match blocks.

Please note that we only match blocks here, because: (a) blocks give a sketch of large class hierarchies, matching them is helpful for users to understand the correspondence between two large class hierarchies; and (b) the sizes of matched block pairs are usually small enough to take current ontology matching techniques to generate accurate 1:1 matching.

4.1 Relatedness Between Blocks Via Anchors

Predefined matched class pairs, called *anchors*, are utilized to find relatedness between blocks. The anchors can be defined by some simple approaches or by experts. For example, the following steps are taken to gain the anchors in our experiments. Please note that the trade-off between the correctness and the number of the anchors should be considered.

1. Find a set of high precision matched class pairs as starting points. This could be done with some string comparison techniques, e.g., [12].
2. Manually remove some incorrect matched class pairs.
3. Manually add some omissions.

Then, the relatedness between blocks can be computed via the anchors gained above. The background idea is that the more anchors we have found between the two blocks, the more related the two blocks are.

Definition 5 (Relatedness between Blocks via Anchors). *Let B_i be a block in class hierarchy H while $B_j^{'}$ be a block in another class hierarchy $H^{'}$. k denotes the number of the blocks in H, and $k^{'}$ denotes the number of the blocks* *in* $H^{'}$. $anchors(B_u, B_v^{'})$ *returns the number of predefined anchors between* B_u *and* $B_v^{'}$. *The relatedness between* B_i *and* $B_j^{'}$ *is defined as follows:*

$$rel_a(B_i, B_j^{'}) = \frac{2 \cdot anchors(B_i, B_j^{'})}{\sum_{u=1}^{k} anchors(B_u, B_j^{'}) + \sum_{v=1}^{k^{'}} anchors(B_i, B_v^{'})}. \tag{6}$$

4.2 Relatedness Between Blocks Via Virtual Documents

Relatedness between blocks are also computed via virtual documents, together with the prevalent TF/IDF [10] technique. In [9], the virtual documents are constructed for concepts (classes, properties or instances) from two ontologies. In this paper, the virtual document of a block is an aggregation of the virtual documents of the classes contained in the block.

The virtual document of a block consists of a collection of weighted tokens, which originate from the local descriptions (e.g., local names) of all the classes it contains and incorporate a weighting scheme to reflect the importance of information appeared in different categories (e.g., tokens appeared in *rdfs:label* are more important than those appeared in *rdfs:comment*). These weighted tokens can be used to reflect the intended meaning of the block.

Then, the virtual document of each block can be represented as a vector in the vector space. The components of the vector are the scores from corresponding tokens, which reflect the relatedness between tokens and the block. The higher the score is, the more the token is related to the block. In addition to the selection of tokens to represent the block, it is common to associate a weight to each token in a block to reflect the importance of that token. Thus, TF/IDF technique is adopted to optimize the vector representation.

Definition 6 (Relatedness between Blocks via Virtual Documents). *Let* B_i *be a block in class hierarchy* H *while* $B_j^{'}$ *be a block in another class hierarchy* $H^{'}$. s_{ik} *denotes the score of a unique token* t_{ik} *in* B_i, *and* $s_{jk}^{'}$ *denotes the score of a unique token* $t_{jk}^{'}$ *in* $B_j^{'}$. D *is the dimension of the vector space. The relatedness between* B_i *and* $B_j^{'}$ *is measured by the cosine value between two vectors:*

$$rel_v(B_i, B_j^{'}) = \frac{\sum_{k=1}^{D} s_{ik} s_{jk}^{'}}{\sqrt{\sum_{k=1}^{D} (s_{ik})^2 \cdot \sum_{k=1}^{D} (s_{jk}^{'})^2}}. \tag{7}$$

If the vectors of B_i, $B_j^{'}$ don't share any tokens, the relatedness will be 0.0; if all the token scores equal completely, it will be 1.0. The score of a unique token t_k in a specific block is defined as follows:

$$\begin{aligned} score(t_k) &= tf \cdot idf \\ &= \frac{t}{T} \cdot \frac{1}{2}(1 + \log_2 \frac{N}{n}), \end{aligned} \tag{8}$$

where t denotes the refined token occurrence, T denotes the total refined occurrence among all the tokens in a specific block, n denotes the number of the blocks containing this token, and N denotes the number of all the blocks.

4.3 Combination

The two kinds of relatedness between blocks computed above are combined to form the overall relatedness. The background assumption is the overall relatedness between two blocks is higher than the one between two other blocks, then it means that the former matched block pairs have more in common than the latter ones.

Definition 7 (Overall Relatedness between Blocks). *Let B_i be a block in class hierarchy H while $B_j^{'}$ be a block in another class hierarchy $H^{'}$. The overall relatedness between B_i and $B_j^{'}$ is defined as follows:*

$$rel(B_i, B_j^{'}) = \beta \cdot rel_a(B_i, B_j^{'}) + (1 - \beta) \cdot rel_v(B_i, B_j^{'}), \tag{9}$$

where $\beta \in [0, 1]$.

Finally, after combining the relatedness between all the blocks from two hierarchies, we select the matched block pairs whose overall relatedness are larger than a given threshold ϵ_2.

5 Experimental Results

In this section, we present some preliminary experimental results in order to evaluate the performance of the partition-based block matching method. To the best of our knowledge, no existing work has shown the experimental results on matching the blocks of large class hierarchies, so we cannot make an objective comparison and measurement. Although manually observing these results is tedious and error-prone, we still believe the evaluation is essential to make progress in this difficult problem.

We apply a pairwise large class hierarchies available in OWL – two Web directory structures proposed in [1] (the data set can be downloaded from OAEI 2005[5]) – to evaluate the performance of our method. Due to lack of space, we don't list the classes contained in each block, the complete results of all the experiments can be found at our Web site that accompanies this paper[6].

5.1 Labeling Blocks

Before introducing the experimental results, it is helpful to label the blocks by representative classes to assist the evaluation. We derive from the descriptions of the most important classes to display the blocks for human observation and understanding. Here, we simply compute the importance of each class based on the size of its children in the block as well as its relative depth.

Definition 8 (Importance). *Let B_i be a block. c_i is a class in B_i, and C_i^{child} is the set of c_i's children which are also in B_i. $|C_i^{child}|$ returns the number of*

[5] http://oaei.inrialpes.fr/2005/directory/

[6] http://xobjects.seu.edu.cn/project/falcon/matching/

classes in C_i^{child}. $depthOf(c_i)$ *returns the depth of class* c_i *in the class hierarchy. The importance of* c_i *in* B_i *is defined as follows:*

$$importance(c_i) = |C_i^{child}| - 2^{relativeDepth(c_i)}, \tag{10}$$

$$relativeDepth(c_i) = depthOf(c_i) - min_{c_k \in B_i}(depthOf(c_k)), \tag{11}$$

where the first part of $importance(c_i)$ *gives more importance to the classes with more children; while the second part gives less importance to the classes deeper in the class hierarchy.*

After computing the importance of all the classes in a specific block, the descriptions of the most important class is selected for the purpose of displaying the block.

5.2 Web Directory Structures

The data set are constructed from Google, Yahoo and Looksmart Web directories as described in [1]. The data set contains 2265 pairwise ontology files and each file contains a path of *rdfs:subClassOf* relations from the leaf class to the root. In our experiments, we firstly merge these 2265 pairwise files to two large class hierarchies, namely the source ontology and the target ontology, and then we apply the partition-based block matching method to them. Experiments are carried out on a 2.8GHz Pentium 4 with 512MB of RAM running on Windows XP Service Pack 2. The parameters used in the experiments are as follows: α in Equation (4) is 0.5, β in Equation (9) is 0.5, ϵ_1 for links filtering is 0, and ϵ_2 for selecting matched block pairs is 0.15.

As depicted in Figure 1, the process of the partition-based block matching method can be divided into three steps. The first step is preprocessing, including loading two large class hierarchies, parsing them and generating links. The next involves partitioning the two hierarchies into blocks, which are then saved to disk. The final step is matching blocks by combining the two kinds of relatedness found via anchors as well as virtual documents. Table 2 gives a breakdown of how long various steps of the matching process take.

Table 2. The runtime per step

	preprocessing	partitioning	matching blocks
time	22s	4s	18s

By preprocessing the two large class hierarchies, the source ontology contains 1067 classes, the number of *rdfs:subClassOf* relations is 1313, the maximum depth is 10, and the number of links is 4063; the target ontology contains 1560 classes, the number of *rdfs:subClassOf* relations is 2331, the maximum depth is 9, and the number of links is 6921. In the partitioning step, the source ontology is partitioned into 6 blocks, the maximum size of the blocks is 204, the minimum size is 142 and the average size is 178; while the target ontology is partitioned

Table 3. The summary of the experimental results

name	classes	subClassOf	depth	links	blocks	anchors	pairs
source	1067	1313	10	4063	6	521	9
target	1560	2331	9	6921	7		

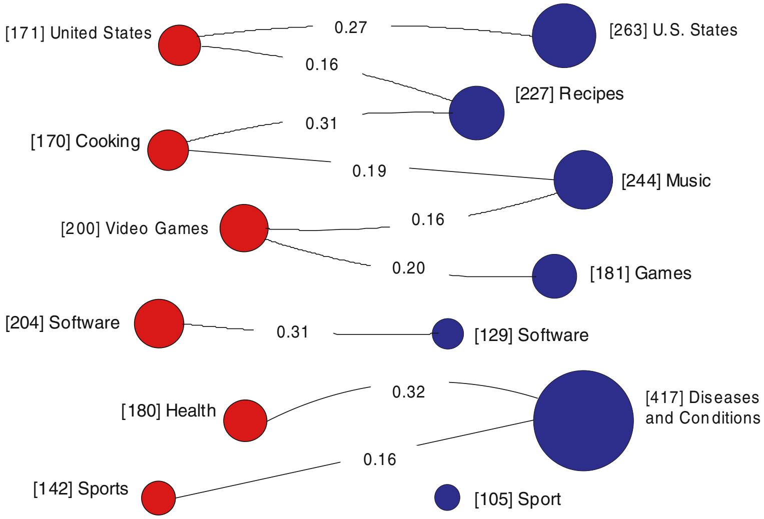


Fig. 2. The details of the experimental results

into 7 blocks, the maximum size of the blocks is 417, the minimum size is 105 and the average size is 223. Finally, in the step of matching blocks, 512 anchors are found by [12], the two kinds of relatedness between blocks are computed via the found anchors as well as virtual documents, and they are combined to gain 9 matched block pairs. The summary of the experimental results on Web directory structures is shown in Table 3.

The graph depicted in Figure 2 shows some useful details of the experimental results. The cycles at the left side represent the blocks of the source ontology and the cycles at the right side represent the blocks of the target ontology. The size of each cycle reflects the number of classes the block contains. The value on each arc shows the overall relatedness between the two matched block pairs.

Discussion. (a) The complete process of the partition-based block matching method takes 44s to complete. Half of the time is spent for preprocessing the two large class hierarchies. It can also be observed that blocks from large class hierarchies can be partitioned with good computational efficiency; (b) 9 matched block pairs are found, 5 matched block pairs are exactly correct by manually evaluating; while 1 potential matched block pairs is missing (Sports vs. Sport), because the number of the anchors from these two blocks is few and the relatedness found via virtual documents is also low due to lack of common tokens. So the approximate precision of our results is 0.56 (5/9) and the recall is 0.83 (5/6); and

(c) current ontology matching techniques can be applied to the matched block pairs for generating 1:1 matched class pairs, for example, we apply V-Doc [9] to the 9 matched block pairs, and then find 798 1:1 matched class pairs.

6 Conclusion and Future Work

In this paper, we propose a method for partition-based block matching that is practically applicable to large class hierarchies. The main contributions of this paper are as follows:

- We present a partitioning algorithm based on both structural affinities and linguistic similarities. The partitioning algorithm is efficient for large class hierarchies, and the time complexity is $O(n^2)$.
- We introduce an approach to matching blocks, which selects matched block pairs by combining the two kinds of relatedness found via predefined anchors as well as virtual documents.
- We describe some preliminary experiments to demonstrate that the partition-based block matching method performs well on our test cases derived from Web directory structures.

As the next step, we are planning to make a comparison between our partitioning algorithm and some others, and the comparison includes both effectiveness and efficiency. Another direction of future research is extending the scope of our method to large-scale ontologies, including both classes and properties. The third direction is how to co-partition (co-clustering) two ontologies, this issue has not yet been touched in the field of ontology matching.

Acknowledgements

The work is supported in part by the NSFC under Grant 60573083, and in part by the JSNSF under Grant BK2003001. The third author of this paper is also supported by NCET (New Century Excellent Talents in University) Program under Grant NCET-04-0472. We would like to thank Ningsheng Jian and Dongdong Zheng for their work in the experiments related to this paper. We would also like to thank anonymous reviewers for their helpful suggestions.

References

1. Avesani, P., Giunchiglia, F., and Yatskevich, M.: A Large Scale Taxonomy Mapping Evaluation. Proceedings of the 4th International Semantic Web Conference. (2005) 67–81
2. Castano, S., De Antonellis, V., and De Capitani Di Vimercati, S.: Global Viewing of Heterogeneous Data Sources. IEEE Transactions on Knowledge and Data Engineering. **13(2)** (2001) 277–297

3. Dhamankar, R., Lee, Y., Doan, A. H., Halevy, A., and Domingos, P.: iMAP: Discovering Complex Semantic Matches between Database Schemas. Proceedings of the 23th ACM SIGMOD International Conference on Management of Data. (2004) 383–394
4. Ehrig, M., and Staab, S.: QOM - Quick Ontology Mapping. Proceedings of the 3rd International Semantic Web Conference. (2004) 683–696
5. Euzenat, J., and Valtchev, P.: Similarity-Based Ontology Alignment in OWL-Lite. Proceedings of the 16th European Conference on Artificial Intelligence. (2004) 333–337
6. Grau, B., Parsia, B., Sirin, E., and Kalyanpur, A.: Automatic Partitioning of OWL Ontologies Using ε-Connections. Proceedings of the 2005 International Workshop on Description Logics. (2005)
7. Guha, S., Rastogi, R., and Shim, K.: ROCK: A Robust Clustering Algorithm for Categorical Attributes. Proceedings of the 15th International Conference on Data Engineering. (1999) 512–521
8. Kaufman, L., and Rousseeuw, P.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons. (1990)
9. Qu, Y. Z., Hu, W., and Cheng, G.: Constructing Virtual Documents for Ontology Matching. Proceedings of the 15th International World Wide Web Conference. (2006) 23–31
10. Salton, G., and McGill, M. H.: Introduction to Modern Information Retrieval. McGraw-Hill. (1983)
11. Shvaiko, P., and Euzenat, J.: A Survey of Schema-Based Matching Approaches. Journal on Data Semantics (IV). (2005) 146–171
12. Stoilos, G., Stamou, G., and Kollias, S.: A String Metric for Ontology Alignment. Proceedings of the 4th International Semantic Web Conference. (2005) 623–637
13. Stuckenschmidt, H., and Klein, M.: Structure-Based Partitioning of Large Concept Hierarchies. Proceedings of the 3rd International Semantic Web Conference. (2004) 289–303
14. Tu, K., Xiong, M., Zhang, L., Zhu, H., Zhang, J., and Yu, Y.: Towards Imaging Large-Scale Ontologies for Quick Understanding and Analysis. Proceedings of the 4th International Semantic Web Conference. (2005) 702–715
15. Winkler, W.: The State Record Linkage and Current Research Problems. Technical Report, Statistics of Income Division, Internal Revenue Service Publication. (1999)

Towards Quick Understanding and Analysis of Large-Scale Ontologies

Miao Xiong, YiFan Chen, Hao Zheng, and Yong Yu

APEX Data and Knowledge Management Lab,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, 200240, P.R. China
{xiongmiao, ivan, zhenghao, yyu}@apex.sjtu.edu.cn

Abstract. With the development of semantic web technologies, large and complex ontologies are constructed and applied to many practical applications. In order for users to quickly understand and acquire information from these huge information "oceans", we propose a novel ontology visualization approach accompanied by "anatomies" of classes and properties. With the holistic "imaging", users can both quickly locate the interesting "hot" classes or properties and understand the evolution of the ontology; with the anatomies, they can acquire more detailed information of classes or properties that is arduous to collect by browsing and navigation. Specifically, we produce the ontology's holistic "imaging" which contains a semantic layout on classes and distributions of instances. Additionally, the evolution of the ontology is illustrated by the changes on the "imaging". Furthermore, detailed anatomies of classes and properties, which are enhanced by techniques in database field (e.g. data mining), are ready for users.

1 Introduction

Ontology[1] plays a key role in the Semantic Web. It not only explicitly represents the taxonomy of a domain (classes and properties), but also sometimes stores the domain information as a knowledge base (individuals or instances). With the development of semantic web technologies, more and more ontologies are developed to formalize the conceptualization of different domains. In some applications, such conceptualization is so large and complicated that the resulting ontology will contain hundreds to tens of thousands of classes and properties. For instance, the OWL version of the National Cancer Institute ontology (NCI ontology[2]) contains more than twenty-seven thousand classes and seventy properties. The Gene ontology[3] contains numerous individuals, too. In order for users to quickly understand and hence make good use of such large-scale ontologies,

[1] In this paper, the term *ontology* refers to a knowledge base that includes concepts, relations, instances and instance relations that together model a domain.

[2] http://www.mindswap.org/2003/CancerOntology/

[3] http://www.geneontology.org

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 84–98, 2006.

© Springer-Verlag Berlin Heidelberg 2006

there must be effective ways to present ontologies and facilitate user browsing and searching.

In our previous work [1], some widely used visualization approaches have been analyzed at length and a novel visualization approach has been proposed to attack their deficiencies. The approach presents a large-scale ontology by a holistic "imaging" which is semantically organized for quick understanding of the subject and content. It can work as a complement of current visualization methods such as class hierarchy view to help user understand the ontology. However, there are still some defects in the approach. In this paper, we try to overcome these defects and add some new features to the approach.

The first improvement is the saving of rendering time. Training the Self-Organizing Maps (SOM) [2] will cost minutes of time when the ontology has hundreds or thousands of classes and properties. Therefore, we offer a new fashion which both preserves the function of SOM and reduces rendering time. Another improvement is the better organization of the classes in the "imaging". While the "imaging" shows both parent classes and their child classes as our previous work, the improved approach preserves hierarchy information of the ontology by placing all sub-classes in their parent class's area.

A new feature introduced to the approach is upon ontology evolution [3,4]. Ontologies are often modified (e.g. errors corrected, more concepts added, or new instances of the domain enriched). In addition, large-scale standardized ontologies are often developed by several researchers collaboratively [5]. For example, the Gene Ontology is maintained by the GO Consortium and the NCI ontology is updated every month. Consequently, in order for users to understand and modify them, a full comprehension of new modifications is indispensable. In other words, evolution of the ontology should be easily noticed and understood by all participants. However, assessing and comprehending changes between different versions of ontologies have been a headachy problem within the ontology community for some time. While most of existing ontology visualization approaches do little on quick understanding of the universal effect of changes, our approach aims to meet these requirements by the enhancement of the imaging which depicts the distinctions between different versions in a holistic view.

We go a step further to help user quickly understand the large-scale ontology with detailed anatomies on classes and properties. Although with a holistic view users can easily find the "hot" classes and comprehend the changes of the ontology, they still need to navigate the ontology to acquire information to answer questions such as "*What food would university students from Shanghai like?*". The answers to such kinds of questions may provide users with additional hidden information of the ontology, especially large-scale ontology. However, it is arduous for users to seek the answers in the huge information "ocean". As techniques in database field such as data mining [6], are widely studied and ready to use, we employ them to enhance the data analysis at instance level to provide users with information of selected class or property. While analysis of statistical information on individuals of ontology are rarely touched by current approaches, our approach provides detailed anatomies of classes and properties at instance level to help users understand specific class and property.

The rest of this paper is organized as follows. Section 2 and Section 3 introduce our previous work and some improvements separately. Section 4 presents the visualization of ontology evolution with examples. Section 5 demonstrates the anatomies on classes and properties by a few user scenarios. Section 6 elaborates the evaluation of our approach. Finally we discuss the related work in section 7 and conclude the paper in section 8.

2 A Brief Introduction of Our Previous Work

Our previous approach [1] produces a holistic "imaging" which not only contains a semantic layout of the ontology classes but also depicts the distributions of the ontology instances and instance relations. In order to lay classes out in an Euclidean space with semantic meaning, we first calculate the semantic connections between them. The semantic connection between two classes takes into account the weighted average of class hierarchy, features of the classes (e.g. parts and attributes), and the number of properties defined between them. Users can customize the weights and set a threshold to discard weak connections. Then, FDP [7] algorithm is chosen to lay the classes out in a 2D plane, with strong connections implying small distances and vice versa. For similar classes are clustered in a small area and large areas are occupied by few classes, we distort the layout using SOM to alleviate the class labeling/annotation problem. After the process, each class occupies an area in the image and owns a point (i.e. the winner neuron [2]). In order to label large-scale ontologies in a screen with a limited size, we compute the importance of each class and only label the most important classes. Since a class is represented by an area, we fill the area with a corresponding color to represent instance numbers of the class. Instance relations are represented with lines between the two corresponding classes' winner neuron points. Just like visualizing the instance distribution, color of lines indicate the number of instance pairs that have the relation.

In user interface aspect, the approach is implemented as a toolkit to support ontology navigation, ontology retrieval, and shallow instance analysis. In the holistic "imaging" of the ontology, neighboring classes typically have stronger semantic connections, and this facilitates user understanding of the ontology. Multiple level-of-detail is presented to help users navigate large-scale ontologies. Finding a certain class is easier than in a mere list because users can get the cluster of the class and then gradually narrow searching scope with level-of-detail techniques. Since "hot" classes and properties are represented with corresponding colors, users can quickly analyze the ontology at instance level. As "hot" classes or properties reveal some information of the ontology, it can help the construction and validation of ontologies. E.g., it may be necessary for ontology engineers to improve the ontology when one class is very "hot" and the others have no instances.

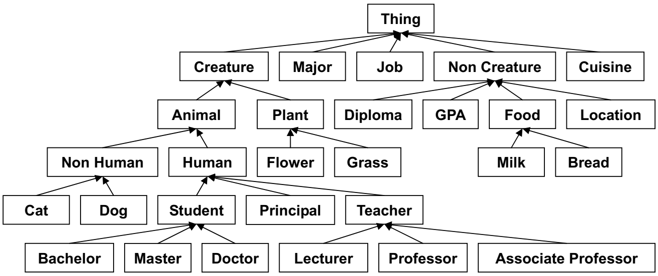


Fig. 1. The class hierarchy of the university ontology. (The properties are omitted. In the following sections, it will be used to depict our approach.)

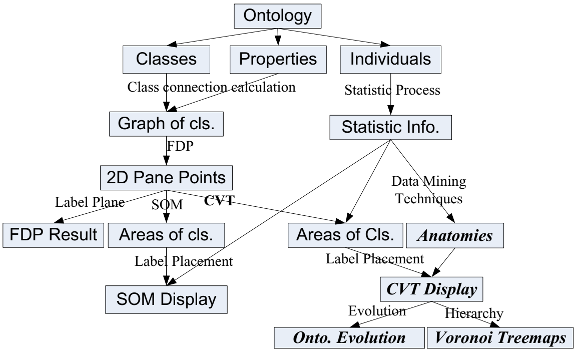


Fig. 2. The data flow of our approach. (The *italic* parts are new features.)

3 Improvements to Previous Work

The improvements (Figure 2) include employing Centroidal Voronoi Tessellation (CVT) [8] and Voronoi Treemaps [9] to enhance the approach's usability and content understandability. While SOM algorithm may achieve an appealing result, it is computationally expensive and requires minutes when the ontology scale is medium or large(e.g. more than 300 classes). This makes the technique less suitable for real-time visualization and exploration of large-scale ontologies. Therefore, we try to alleviate the problem by using CVT as a complement visualization method of SOM when it is necessary to finish the rendering in much less time. In addition, Voronoi Treemaps is introduced to show hierarchical information of ontologies.

3.1 CVT

Given N points in a plane, Voronoi diagram is the partitioning result of the plane that every area is a convex polygon containing exactly one generating point. Every point in the polygon is closer to its generating point than to any other. The defect of the Voronoi diagram is that it is sometimes hard to read

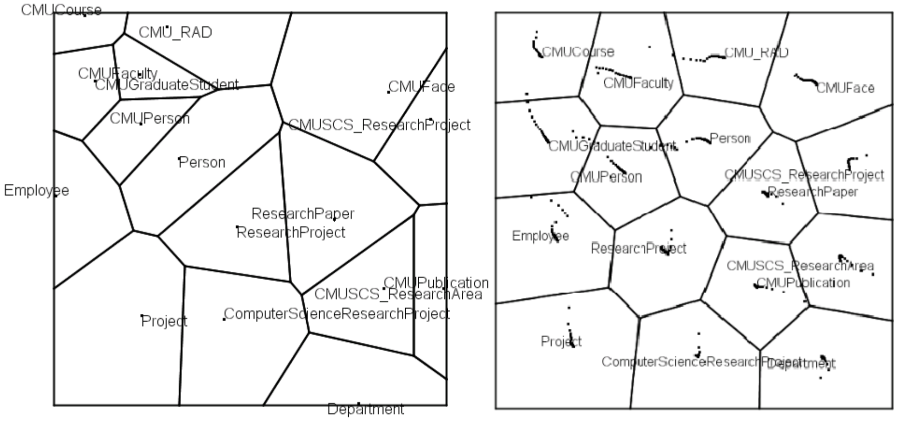


Fig. 3. CVT result after adjustment of Vonoroi diagram. (The traces are the movements of centroids. The ontology used in this figure is CMU schema part of the TAP ontology.)

the diagram for some polygons occupy a large area and some polygons clusters in a small area (Figure 3). By adjusting the original points, we can get almost same area while preserving the neighboring character of the diagram. CVT [8] is such an adjustment that generating points are centroids (centers of mass) of the corresponding polygons. It consists of two basic steps: 1) draw Voronoi diagram according to some points; 2)adjust each point into the mass center of its polygon. The two steps repeat iteratively until the error between all point positions and the mass centers of their Voronoi polygons is below a given ε. The adjustment process of CVT is depicted with a part of TAP ontology in Figure 3.

3.2 Result Compare Between SOM and CVT

To do a compare of CVT and SOM, we render the same ontology (university ontology) respectively. Figure 4 shows the placement of CVT and SOM. It is easy to find the similar and different characters between CVT and SOM from Figure 4:

- Both of them enable the image to be readable by adjusting results of FDP.
- Both of them preserve the neighboring character of classes with strong semantic connections. For example, the neighboring order of **Plant**, **Flower**, and **Grass** are preserved in the two images.
- As CVT is composed by polygons, some neighboring classes may lose the neighboring character as depicted with SOM. For example, while **Lecturer** and **Teacher** are close to each other in the SOM, they are totally separated by **Doctor** in CVT.

While the time complexity of CVT is $O(NlogN)$, where N stands for the number of the classes, that of SOM is $O(M^2)$, where M is the neuron number (the

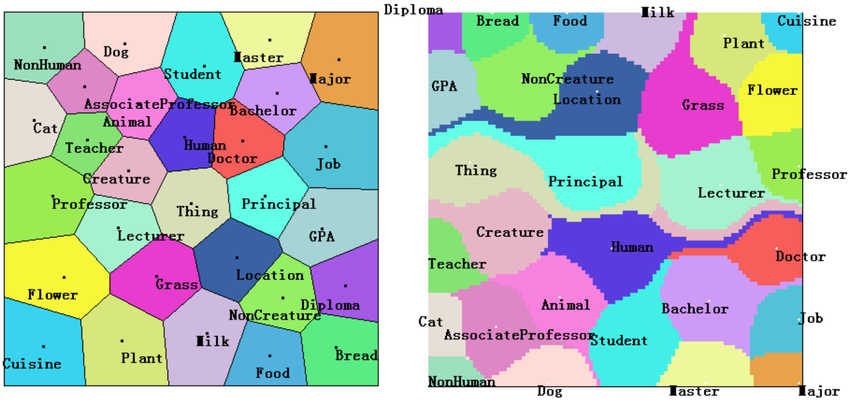


Fig. 4. Compare of CVT and SOM. (The left image is the result of CVT; the right one is the result of SOM.)

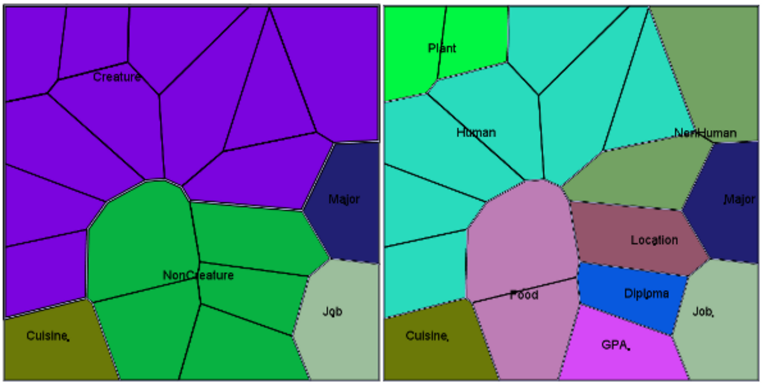


Fig. 5. Result of the revised Voronoi treemap

resolution number of the image). For large ontologies, low resolution is far from satisfactory; and the rendering time of SOM will become unacceptable when the scale of SOM becomes larger. Therefore, we decide to introduce CVT to facilitate the visualization of large-scale ontologies as complement of SOM.

3.3 Revised Voronoi Treemap

Voronoi treemaps [9] combines CVT and Treemap [10]. Contrary to existing rectangle-based Treemap layout algorithms, its layout algorithm, which is composed of polygons and not necessary to be rectangle, can show hierarchy information. In order to both reserve the semantic layout(i.e. the classes with strong semantic connection are close to each other) and hierarchy information (i.e. all sub-classes are put in their parent's area), we revise the Voronoi Treemaps algorithm's calculation sequence. In other words, while they usually split the whole

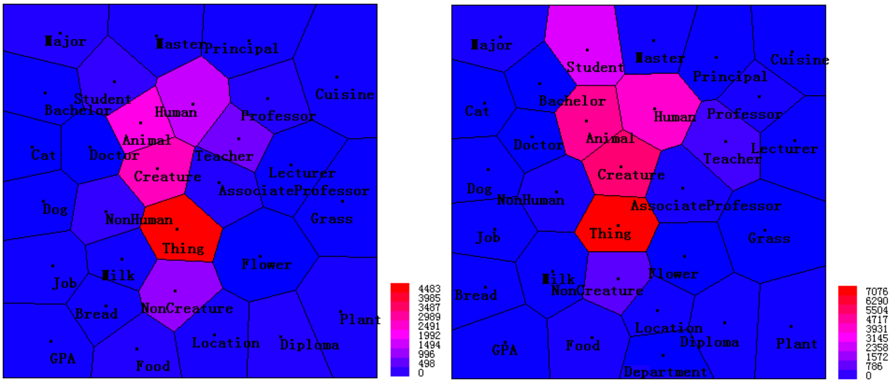


Fig. 6. Ontology evolution depicted by CVT

area into several small areas and then split these small areas independently in Voronoi Treemaps (top-down), we first calculate areas of all leaf nodes and then assign each non-leaf node (parents) with the areas of its sub-classes (bottom-up). As ontologies may contain class that has multiple super-classes, we split the area into several pieces and assign each piece to its nearest super-class in the "imaging" (when the sup-sub classes are too complex to be divided and assigned, our algorithm may show inexact result, too). The result of Voronoi Treemaps of the university ontology is shown in Figure 5. The left part is from the highest viewport, only the direct subclasses of the "root" could be seen. The right one depicts the case that covered subclasses are revealed when the viewport goes down.

4 Ontology Evolution

Since browsing and checking changes on classes, properties, and individuals one by one are time-consuming and listing all the changes of the ontology is also awkward when the changes are innumerable, we demonstrate ontology evolution by different holistic "imagings". The visualization method is introduced as the complement of the accurate presentation (e.g all differences are listed in a view).The visualization of ontology evolution aims to help users in collaborative construction of large-scale ontologies. As differences between versions are depicted by the changes on the holistic "imaging" enhanced by semantic connections, users can quickly find and comprehend the changes. The evolution of ontology includes three parts, i.e. classes, properties, and individuals. Two different versions of the evolving university ontology are depicted in Figure 6. The university ontology was not complete when doing the evolution visualization.

It is easy to find from Figure 6 that: 1)a new class **Department** is added as sub-class of **NonCreature**; 2) the instance number of class **Student** increases. The corresponding changes of the visualization "imaging" are: 1)classes neighboring **NonCreature** are distorted and moved to nearby areas as the new class

Department will occupy the area near **NonCreature**. Therefore, the changes of the ontology can be noticed by the distortion of classes in the “imaging”. 2)The color depth of **Student** becomes “hotter” for new instances are filled.

5 Anatomies of Class and Property

While quickly navigating and searching “hot” classes and properties are important for understanding ontologies, acquiring detailed information on them is also necessary for further investigation. Facing an ontology with numerous instances, people are often at a loss in the huge information “ocean”. Although we have proposed an innovative method to highlight the “hot” classes and actively interconnected classes in our previous work, it is still hard for users to understand the ontology in detail. Considering the following questions on the *university ontology*:

1. Does the university focus on research or engineering?
2. When a student likes to eat bread, does s/he like to drink milk?
3. What are the common characters of elite students with major computer science?
4. For a businessman who wants to run a restaurant around university in Shanghai, which kind of cuisine he should provide to make the business succeed?

In this paper, we go a step further to provide users with more information acquired at instance level to quickly deal with these questions. As data mining techniques have been widely studied in database field and have been integrated in some commercial databases, they are introduced to anatomize classes and properties in our approach. The important and interesting fact for data mining is that it can discover information hidden in knowledge bases. As large-scale ontology is a huge knowledge base which stores numerous instances of the domain, we try to discover helpful information with these techniques. Consequently, our approach provides users with more information underlying the ontology knowledge base. In other words, it presents users with analysis at instance level about 1)anatomy of class i)constitution of class ii)association rule between properties, and 2)anatomy of property.

5.1 Anatomy of Class

Class is one of most important concepts in the ontology. By the navigation method provided by our previous work, users can conveniently locate the needed class. But it is still hard for them to acquire statistical information of classes with numerous instances. Providing users with statistic information of the class may help them a lot for quick understanding large-scale ontologies. Therefore, we adopt data mining techniques to assist the discovery of information through instance level ontology analysis. With the instance patterns extracted from the ontology knowledge base, the constitution of class and association rule between properties are presented to users for quick understanding and analysis of the ontology.

Table 1. The constitution of **Student** and **Teacher** in the university ontology

Type	*Count*	*Percentage*
Student	18562	100%
Bachelor	7560	40.7%
Master	8156	43.9%
Doctor	2846	15.3%

Type	*Count*	Percentage
Teacher	1506	100%
Lecturer	564	37.5%
Associate-Professor	423	28.1%
Professor	519	34.5%

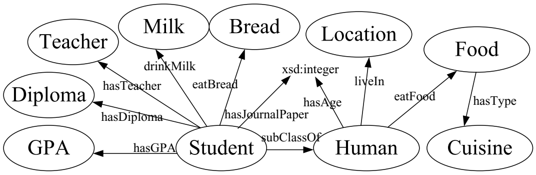


Fig. 7. Snippet of the university ontology

Constitution of Class. Taking for example the university ontology shown in Figure 1. As to the first question proposed at the beginning of Section 5, our approach provides user with the constitution of focused class. By simple clicks on **student** and **teacher**, which are classes in the ontology, the related information are collected and presented to the user. The following tables are the constitutions of **student** class and **teacher** class of the ontology. With the constitution listed in Table 1, the user can have an intuition that the university focuses more on research rather than engineering as most of the students are graduate students and most of teachers are professors or associate professors.

Association Rule between Properties. Association rule mining is another deep studied technique in database field and can be applied to the field of ontology. Taking the university ontology for example, in which the **student** class and the two properties, namely **eatBread** and **drinkMilk** (Figure 7), are defined, we are able to answer the second question proposed at the beginning of Section 5. By selecting the property **eatBread** and **drinkMilk** respectively, our approach will calculate the association rules (Table 2). The results shows that: 1)when a student likes to eat bread, the probability of liking to drink milk is 86.9 percents; 2)when a student likes to drink milk, the probability of liking to eat bread is 93.2 percentage. With the information collected in Table 2, users may easily calculate support and confidence of the association rule[4].

In addition, our approach provides users with ability to add restrictions on multiple properties. For example, we can get the following association among **hasGPA**, **hasJournalPaper**, and **hasDiploma**:

$$((\textbf{hasGPA}\text{=“HighGPA”})\wedge(\textbf{hasJournalPaper}>1))\Rightarrow(\textbf{hasDiploma}\text{=“Doctor”}).$$

[4] Though the property **hasGPA**, **hasTeacher** are 100% in each case, they are not suitable to be treated as having association with **eatBread** or **drinkMilk** as every student in the ontology has GPA and teacher.

Table 2. The association rules related to **eatBread** and **drinkMilk**

Property	*Count*	*Percentage*	*Property*	*Count*	Percentage
eatBread	14562	100%	drinkMilk	13264	100%
hasGPA	14562	100%	hasGPA	13264	100%
hasTeacher	14562	100%	hasTeacher	13264	100%
hasDiploma	2846	19.5%	hasDiploma	2573	19.4%
drinkMilk	12654	86.9%	eatBread	12362	93.2%

As the university ontology reveals, a student whose GPA is "HighGPA" and has published more than one journal paper has probably got a doctor diploma. The answer of the third question proposed at the beginning of Section 5 can also be retrieved by constraining the properties **hasMajor**="CS" and **hasGPA**="HighGPA".

5.2 Anatomy of Property

Property plays as important, on some occasion even more, a role than class in ontology. In order for users to have good comprehension on some specific property, our approach allows them to add some restrictions on its domain. By a convenient statistic operation, individuals' common features (i.e. the common value of some property) of the range class are retrieved. As the fourth question proposed at the beginning of Section 5, the businessman who want to run a restaurant may focus on the property **eatFood** and the triple <**Human**, **eatFood**, **Food**> rather than any single class. He may find something useful by restricting the location of **Human** to **Shanghai**. Finding that nearly half of the students have location in **Shanghai** like to eat sweet food, which conforms to the widely accepted viewpoint in China that "*people living in Shanghai like to eat sweet food*", he may run a restaurant offering sweet cuisine. In addition, restrictions are allowed on multiple properties, e.g. the businessman can further restrict the **hasAge** to larger than **20** to satisfy specific needs.

6 Evaluation of the Approach

In order to get a comprehensive understanding of our approach, we conducted an evaluation which consists of the following 3 aspects:

- Rendering speed.
- Time for an ordinary user to learn the visualization method.
- Understandability of the approach's result, or content understandability.

No matter the ontology is RDF or OWL, our approach are able to be implemented in any ontology visualization toolkit. We integrated it in the Orient[5] [11] system as our validation testbed. Three ontologies, from medium to large scale: Universtity ontology, SWETO[6], and TAP[7] ontology (Table 3) are run for mea-

[5] http://apex.sjtu.edu.cn/projects/orient/

[6] http://lsdis.cs.uga.edu/Projects/SemDis/sweto/

[7] http://tap.stanford.edu/tap/download.html

Table 3. Rendering speed of CVT and SOM. (As the sizes of SOM were the same, their renderings cost almost the same time. However, the size of SOM is also decided by the scale of the ontology and when the size of SOM becomes bigger, the rendering will cost much more time.)

Ontology	*Concept No.*	*Property No.*	*Individual No.*	*SOM(second)*	*CVT(second)*
SWETO	116	58	12563	39	8
TAP Place	76	8	3639	28	3
University	29	11	31252	29	0.4

suring the rendering speed and understandability. Different versions of university ontology is rendered to depict ontology evolution. The experiments were performed with a PC with P4 2.4GHz CPU and 1G Memory.

Participants. The validation was conducted by the authors and other 16 computer science students on semantic web from Shanghai Jiao Tong University. The participants, which includes 4 under-graduate students, 10 graduate students, and 2 doctor candidates. Participants were voluntary to join the user study without financial incentives.

Learning Process. To arm users with basic knowledge of our visualization approach, we gave a brief introduction (about ten minutes) on the techniques used in our approach. After the introduction, participants began to use the visualization system and evaluate our approach.

Validation. Participants were not familiar with the three ontologies at the beginning. The following tasks are performed by them to evaluate the understandability of visualization result with our approach:

1. Wait for the result of our visualization approach.
2. Find the “hot” classes and properties of the ontologies.
3. Describe main topic of the ontology.
4. Tell the differences between two versions of ontologies with evolution visualization result.
5. Give comments on knowledge discovered from the ontologies.
6. Discover the utility of anatomies on classes and properties.

The following questions on different aspects of the approach are asked after their using of the tool.

1. Is the approach necessary for understanding large-scale ontologies?
2. Does the visualization techniques achieve the desired result?
3. How about finding the interesting classes and properties?
4. How many time you spend in understanding the result?
5. How much has the ontology changed between versions?
6. How would you summarize the changes that have occurred with respect to their type?
7. What and where have the changes occurred in the ontology?
8. How about the anatomies in the approach?

Feedback. We collected the feedbacks from participants and summarized as follows:

- All participants think that the speed of CVT is acceptable (16/16); Seven participants think that the speed of SOM is too slow (7/16) and others think that the time less than 1 minutes are tolerable.
- All participants think that it is necessary to have a quick understanding of large-scale ontologies. After about ten minutes' short introduction on our approach, participants are able to judge the effect of visualization results.
- All participants are able to retrieve the "hot" classes and properties of the ontology in less than 2 seconds even the ontology has more than 100 classes.
- All participants are able to locate the differences of two versions of the same ontology in less than 5 seconds; but 3 participants complain that the searching time is too long. Some complains that finding evolution caused by property is not as convenient as class. They all have a quick understanding of the changes and can tell us the main differences.
- Four participants do not understand the result of CVT as well as SOM. But all participants think that it is enough for quick understanding, which is the complement of accurate and detail understanding.
- All participants think that the anatomies provide them with convenience to understand the ontology at instance level. But 3 participants complains about the holdup(usually less than two seconds) caused by real-time computing.
- All participants think that it is a pity for the current system to be hard to scale to some huge ontologies (e.g. the NCI Thesaurus with more than 20,000 classes).

7 Related Work

A number of works have contributed to facilitate the understanding of ontologies. Ontology visualization is the most intuitive way and a lot of visualization works (e.g. OntoViz [12], TGVizTab [13], Jambalaya [14]) have contributed to precise representations of ontology schema. Our approach is different from these ones as it presents a holistic view which is applicable to large-scale ontologies with level-of-detail. Jambalaya depicts the ontology hierarchy information with level-of-detail by putting sub-classes in their parent's area. Our approach preserves the hierarchy information by employing the Voronoi Treemaps; in addition, it also arranges classes with semantic connections layout.

Currently, most of ontology visualization approaches do not support well on the evolution of large-scale ontologies. PromptViz [15] is a plug-in based on Protégé[8] which represents differences between two versions of an ontology annotated in treemap layouts. When a difference is selected, a detailed list of changes is presented to the user. Our approach is different for it emphasizes the impacts of changes to the entire ontology.

[8] http://protege.stanford.edu/

Analysis of an ontology at instance level has not been involved in most ontology related tools currently. Our approach, however, manages to visualize the distribution of ontology instances and instance relations. There are also some other ontology visualization methods that aim to integrate instance information into the visualization. The Spectacle system [16] is designed to present a large number of instances with a simple ontology to users. Each instance is placed into a cluster based on its class membership. Instances which are members of multiple classes are placed in overlapping clusters. This visualization approach provides a clear and intuitive depiction of the relationships between instances and classes. Our approach can not visualize the instance overlap like Spectacle, which is the cost of our choosing to present a holistic view of large-scale ontologies. Peter et al. [17] have used a genome-wide GO-based data mining approach to examine potential correlations between the GO terms and the frequency of 50 CpG islands in genes annotated to those terms. Jill Cheng et al. [18] have introduced the NetAffx Gene Ontology Mining Tool which presents visualization combining biological annotation with expression data, encompassing thousands of genes in one interactive view.

The visualization techniques and algorithms employed in our approach are also widely introduced in other knowledge visualization fields. Infosky [19] is a system that tries to enable the exploration of large and hierarchically structured knowledge spaces, and it presents a two-dimensional graphical representation with variable magnification, much like providing users a real-world telescope. One of the key algorithms employed in InfoSky [19] is a spring-embedding algorithm, which is used to cluster documents or document collections. WEBSOM [20] is developed for visualizing document clusters based on the SOM. Similar documents become close to each other on the map, like books on shelves in a well-organized library. Some other systems, like Themescape by Cartia Inc. and ET-Map [21], also visualize large numbers of documents or websites using the SOM algorithm. CVT is used introduced by Du et al. [8] and widely used in the visualization of software metrics [9]. Voronoi Treemaps [9] is a hierarchy-based visualization approach for software metrics. The layouts are based on arbitrary polygons instead of standard rectangle based treemaps and are computed by the iterative relaxation of Voronoi tessellations.

8 Conclusion and Future Work

In this paper we present a novel approach to help users quickly understand large-scale ontologies. Our approach produces a holistic "maging" and anatomies to depict large-scale ontologies. The alternation between CVT and SOM can help user balance rendering time and result. In addition, Voronoi Treemap preserves the hierarchy information, and ontology evolution is also illustrated by changes of the "imaging". Furthermore, detailed anatomies of classes and properties, which are enhanced by techniques in database field(e.g. data mining), may help users further understand on specific classes or properties.

Through a conducted validation process, we watched the resulting user interface and found it facilitates quick understanding and analysis of large-scale ontologies. Although our approach deals well with ontologies with numerous instances, the rendering of large-scale ontologies (e.g. more than 20,000 classes) still costs minutes or even hours of time. Therefore, we are planing to improve the rendering algorithm's speed or employ more faster algorithms into our approach as the next step.

References

1. Tu, K., Xiong, M., Zhang, L., Zhu, H., Zhang, J., Yu, Y.: Towards imaging large-scale ontologies for quick understanding and analysis. In: ISWC. (2005) 702–715
2. Kohonen, T.: Self-Organizing Maps. Springer (1995)
3. Noy, N.F., Klein, M.: Ontology evolution: Not the same as schema evolution. SMI technical report SMI-2002-0926 (2002)
4. Klein, M., Noy, N.F.: A component-based framework for ontology evolution. In: Workshop on Ontologies and Distributed Systems at IJCAI-2003, Acapulco, Mexico (2003)
5. Pinto, H.S.A.N.P., Martins, J.P.: Evolving ontologies in distributed and dynamic settings. In: KR. (2002) 365–374
6. Galiano, F.B., Marín, N.: Data mining: Concepts and techniques - book review. SIGMOD Record **31** (2002) 66–68
7. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. Software - Practice and Experience **21** (1991) 1129–1164
8. Du, Q., Faber, V., Gunzburger, M.: Centroidal voronoi tessellations: Applications and algorithms. Volume 41. (1999) 637–676
9. Balzer, M., Deussen, O., Lewerentz, C.: Voronoi treemaps for the visualization of software metrics. In: SOFTVIS. (2005) 165–172
10. Johnson, B., Shneiderman, B.: Tree maps: A space-filling approach to the visualization of hierarchical information structures. In: IEEE Visualization. (1991) 284–291
11. Zhang, L., Yu, Y., Lu, J., Lin, C., Tu, K., Guo, M., Zhang, Z., Xie, G., Su, Z., Pan, Y.: ORIENT: Integrate ontology engineering into industry tooling environment. In: ISWC. (2004)
12. Sintek, E.: OntoViz: Ontoviz tab: Visualizing protege ontologies. (2003)
13. Alani, H.: TGVizTab: An ontology visualization extension for protege. In: in Knowledge Capture 03 - Workshop on Visualizing Information in Knowledge Engineering, Sanibel Island, FL (2003)
14. Storey, M.A.D., Noy, N.F., Musen, M.A., Best, C., Fergerson, R.W., Ernst, N.: Jambalaya: an interactive environment for exploring ontologies. In: IUI. (2002) 239–239
15. Perrin, D.: Prompt-viz: Ontology version comparison visualizations with treemaps. Master's thesis, University of Victoria, BC, Canada (2004)
16. Fluit, C., Sabou, M., van Harmelen, F.: Ontology-based information visualization. In: Proceedings of Information Visualization '02. (2002)
17. Peter N. Robinson, a.U.B., Lopez, R., Mundlos, S., Nrnberg, P.: Gene-ontology analysis reveals association of tissue-specific 50 cpg-island genes with development and embryogenesis. Human Molecular Genetics **13** (2004) 1969–1978

18. Cheng, J., Sun, S., Tracy, A., Hubbell, E., Morris, J., Valmeekam, V., Kimbrough, A., Cline, M.S., Liu, G., Shigeta, R., Kulp, D., Siani-Rose, M.A.: Netaffx gene ontology mining tool: a visual approach for microarray data analysis. Bioinformatics **20** (2004) 1462–1463
19. Andrews, K., Kienreich, W., Sabol, V., Becker, J., Droschl, G., Kappe, F., Granitzer, M., Auer, P., Tochtermann, K.: The infosky visual explorer: exploiting hierarchical structure and document similarities. Information Visualization **1** (2002) 166–181
20. Kaski, S., Honkela, T., Lagus, K., Kohonen, T.: Websom - self-organizing maps of document collections. Neurocomputing **21** (1998) 101–117
21. Chen, H.C., Schuffels, C., Orwig, R.: Internet categorization and search: A self-organizing approach. Journal of Visual Communication and Image Representation **7** (1996) 88–102

Matching Large Scale Ontology Effectively

Zongjiang Wang, Yinglin Wang, Shensheng Zhang, Ge Shen, and Tao Du

Dept. of Computer Science
Shanghai Jiaotong University, 200030, China
microw@sjtu.edu.cn

Abstract. Ontology matching has played a great role in many well-known applications. It can identify the elements corresponding to each other. At present, with the rapid development of ontology applications, domain ontologies became very large in scale. Solving large scale ontology matching problems is beyond the reach of the existing matching methods. To improve this situation a modularization-based approach (called MOM) was proposed in this paper. It tries to decompose a large matching problem into several smaller ones and use a method to reduce the complexity dramatically. Several large and complex ontologies have been chosen and tested to verify this approach. The results show that the MOM method can significantly reduce the time cost while keeping the high matching accuracy.

1 Introduction

Matching is a critical operation in many well-known application domains such as ontology integration, semantic web, data warehouse, e-commerce, etc. The increasing awareness of the benefits of ontologies for information processing has lead to the creation of a number of such ontologies for real world domains. Many different solutions have been proposed to the matching problem. Examples include Cupid, COMA, Glue, Rondo, and S-Match, etc[1-5]. However, in complex domains such as medicine these ontologies can contain thousands of concepts. The previous approaches were typically applied to small ontologies in which most correspondences could be automatically determined without much difficulty in a reasonable time. However, as surveyed in, most small ontologies are structurally rather simple and of the size of ontology are less than 100 components (classes, properties). Unfortunately, the effectiveness of automatic match techniques studied so far may significantly decrease for larger scale ontologies[2] because larger ontologies increase the likelihood of false matches. To improve this situation a modularization-based approach (called MOM) was proposed in this paper. This approach tries to decompose a large matching problem into several smaller ones and use a method to reduce the complexity dramatically.

The rest of the paper is organized as follows. In section 2, we firstly give a brief introduction of our system architecture, and then describe the components of system in detail. The experiments and evaluation are given in section 3. Finally, we conclude our work with a discussion in section 4.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 99 – 105, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 A Modularization-Based Ontology Matching Approach

We propose a Modularization-based Ontology Matching approach (we call it MOM later). This is a divide-and-conquer strategy which decomposes a large matching problem into smaller sub-problems by matching at the level of ontology modules. As illustrated in Fig.1, the strategy encompasses four steps: (1) partition the large ontologies into suitable modules, (2) identify the most similar modules in two sets of modules, (3) use the OPM algorithm to match two similar modules, and (4) combine the module match results. By reducing the size of the mapping problem we not only can obtain better performance but also can improved match quality compared to previous ontology mapping methods.

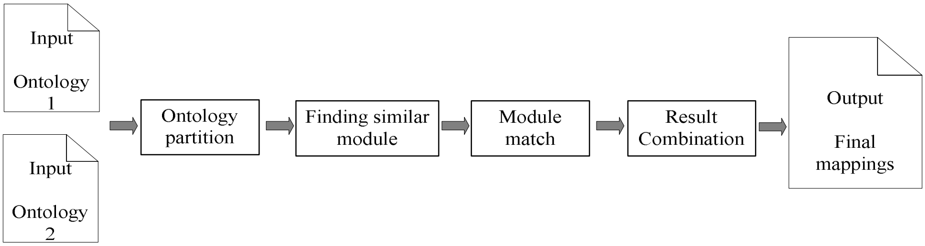


Fig. 1. Matching process in MOM

2.1 Ontology Partition

In this section, we show how to partition the large ontologies into small modules (Fig.2). We take the approach of[6].

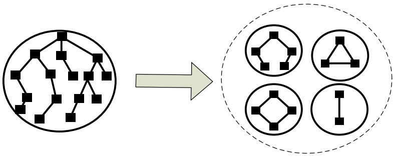


Fig. 2. Ontology Partition

This method takes the E-connection as the theoretical foundation. In a Semantic web context, E-connection contains a set of "E-connected" ontologies. Each of the E-connection is modeling a different application domain, while the E-connection is modeling the union of all domains. For brevity, an E-connection is an extended OWL-DL, which adds the functions to define and use the link property. After introducing a series of definitions, such as semantic encapsulation, strongly encapsulating and module, the authors then try to find the relevant axioms for each entity in the original ontology. The main idea of this approach is to transform the input ontology into an E-connection with the largest possible number of connected knowledge bases and keep the semantics of the original ontology in a specific way.

2.2 Finding Similar Modules

The goal of this step is to identify modules of the two ontologies that are sufficiently similar to be worth matching in more detail. This aims at reducing match overhead by not trying to find correspondences between irrelevant modules of the two ontologies. Assume the first ontology has M modules, and the second one has N modules, and the approach should execute $\mathrm{M} \times \mathrm{N}$ matching. With the help of modules matcher, we remove the irrelevant module-pairs and obtain the L similar module-pairs. Generally, L is much smaller than $\mathrm{M} \times \mathrm{N}$, and this will avoid unnecessary calculation if compared with other methods.

The problem of finding the most similar L module-pairs may be transformed to the problem of finding the maximum bipartite match[7].

Definition: a bipartite graph $\mathrm{G=(X,Y,E)}$ is a simple graph defined as follows:

- X is the set of vertices which denotes a modules of first ontology
- Y is the set of vertices which denotes a modules of second ontology
- E is the set of edges which all go between the X and Y. The weight of the edge is the similarity of the vertices.

The question is to find a match $\mathrm{M} \subseteq E$ such that $\mathrm{w(M)}=\sum_{e \in M} w(e)$ is the maximum.

To solve the maximum bipartite match problem, we use the Hungary arithmetic which can find out the match of bipartite graph and the Kuhn arithmetic which can find out the maximum one based on Hungary arithmetic[7].

Now we introduce how to get the similarity of the two modules. In order to compare two modules (they are parts of the ontologies) and measure the similarity between them, we use the similarity measure $\mathrm{Sim(O_1,O_2)}$ between two ontologies, $\mathrm{O_1}$ and $\mathrm{O_2}$, which is based on two values: (1) lexical similarity and (2) conceptual similarity[8].

2.2.1 Lexical Similarity

We use edit distance method to compare two lexical terms.

$$\mathrm{Sim}(L_i, L_j) = \max(0, \frac{\min(|L_i|,|L_j|) - ed(L_i, L_j)}{\min(|L_i|,|L_j|)}) \in [0,1] \tag{1}$$

where $\overline{\mathrm{L}}_1$ is a lexicon of ontology $\mathrm{O_1}$ which includes a set of terms for ontology concepts L_1^{C}, and a set of terms for ontology relations L_1^{R}. L_i is a term of $\overline{\mathrm{L}}_1$, and L_j is a term of $\overline{\mathrm{L}}_2$. $\mathrm{Sim}(L_{\mathrm{i}}\ ,L_{\mathrm{j}}\ )$ returns a number between 0 and 1, where 1 stands for perfect match and zero for no match. Then we can get the lexical similarity between the two ontologies:

$$\mathrm{Sim}(\overline{\mathrm{L}}_1,\overline{\mathrm{L}}_2) = \frac{1}{|\overline{\mathrm{L}}_1|} \sum_{L_{\mathrm{i}} \in \overline{\mathrm{L}}_1}^{n} \max_{L_{\mathrm{j}} \in \overline{\mathrm{L}}_2} \mathrm{Sim}(L_i, L_j) \tag{2}$$

We notice that $\mathrm{Sim}(\overline{\mathrm{L}}_1,\overline{\mathrm{L}}_2)$ is an asymmetric measure that determines the level to which the lexical level of a sign system $\overline{\mathrm{L}}_1$ (the target) is covered by the one of a second sign system $\overline{L}_2$ (the source). Obviously, $Sim(\overline{L}_1, \overline{L}_2)$ may be quite different from $Sim(\overline{L}_2, \overline{L}_1)$. Let us definite the relative number of hits:

$$SetHit(\overline{L}_1, \overline{L}_2) = \frac{|\overline{L}_1 \cap \overline{L}_2|}{|\overline{L}_1|} \tag{3}$$

To make the $Sim(\overline{L}_1, \overline{L}_2)$ correct in all conditions, we must assure the value of $SetHit(\overline{L}_1, \overline{L}_2)$ is less than 1.

2.2.2 Conceptual Similarity

Conceptually, we may compare semantic structures of ontologies O_1, O_2 that vary for concepts A_1, A_2. In our model the conceptual structures consist of two parts: one is the similarity between the two taxonomies of the ontologies, another is the similarity between the two sets of the relations of the ontologies[8].

2.2.3 Total Similarity

The total similarity between two ontologies is the combination of lexical similarity and conceptual similarity. Here, a fixed weighting scheme is applied for the combination. The weights can be chosen by the expert experience.

$$Sim(O_1, O_2) = W_{lexical} * Sim_{lexical} + W_{conceptual} * Sim_{conceptual} \tag{4}$$

where $W_{lexical} + W_{conceptual} = 1$.

2.3 Module Match

Here we used a matching method OPM (Ontology Parsing graph-based Mapping method)[9]. The algorithm has 5 steps: ontology parsing, ontology parsing graph generation, lexical similarity calculation, similarity iteration, and graph match (Fig 3).

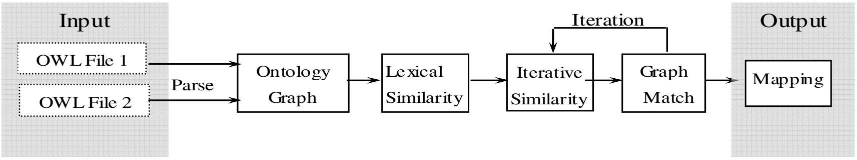


Fig. 3. Architecture of OPM

2.4 Result Combination

Because our task is to determine the match result for two complete ontologies, so the match correspondences for two modules mapping need to be combined with the match result into a complete one.

2.5 Analysis of Run-Time Complexity

Now we discuss the complexity of the MOM. We consider two situations:

(1) For the ontologies that cannot be modularized, the complexity of the partition module is $O(|V|^5)$. Since the complexity of the exact mapping module is

$O(|V|^{5.5}) \sim O(|N|^{5.5})$ (N stand for the number of entities of the OP-graph), therefore, the complexity of the whole algorithm is $O(|N|^{5.5})$, which is same as the complexity without modularizing. (2) For the ontologies that can be modularized, the complexity of the partition module is $MO(|U|^5)$, where $N=MU$, and M is the modules number. From the above, we know the complexity of the module match is $O(|V|^{5.5}) \sim O(|U|^{5.5})$. So the complexity of the MOM algorithm is $MO(|U|)^5 + MO(|U|^{5.5}) \sim MO(|U|^{5.5})$. The complexity of the algorithm without modularization is $O(|N|^{5.5}) = O(|MU|^{5.5}) = O(|M|^{5.5})\ O(|U|^{5.5})$. Comparing the two results, we know the complexity of the whole algorithm decreases by $O(|M|^{4.5})$ after modularizing the large ontologies.

3 Experimental Evaluation

In order to evaluate our approach, we have conducted some experiments. In the experiment, we evaluated MOM on some practical large data sets: web services ontologies, medical ontologies and tourism ontologies. These ontologies are from different places and have 172-646 concepts (see Table 1). The ontologies of each pair are similar to each other.

Table 1. Ontologies in experiments

ontologies		concepts	Properties		Instances number	manual mapping
			Data properties	object properties		
Web services	1	209	8	228	16	171
	2	172	13	122	246	158
Medical	1	398	9	166	163	236
	2	443	13	206	247	251
Tourism	1	549	8	312	262	398
	2	646	21	241	354	407

We use standard information retrieval metrics to evaluate our method and compare with other methods[10].

$$\Pr ec = \frac{|m_a \cap m_m|}{|m_a|}, \mathrm{Re}\, c = \frac{|m_m \cap m_a|}{|m_m|} \tag{5}$$

where m_a are mappings discovered by MOM(or OPM) and m_m are mappings assigned by experts.

We took the OPM as the baseline method to test the effect of MOM.

OPM - It uses the two ontologies as the input, and does not consider the size of the ontologies.

MOM - It focus on the large scale ontology matching problems.

Table 2. Experimental comparison between OPM and MOM

Data set	mapping	OPM		MOM	
		Prec	Rec	Prec	Rec
Web services	1 to 2	76.1	73.2	76.0	73.1
	2 to 1	75.2	71.4	75.8	71.6
Medical	1 to 2	71.1	69.2	70.8	69.0
	2 to 1	76.7	74.2	76.8	74.1
Tourism	1 to 2	80.3	73.6	79.1	75.6
	2 to 1	78.5	73.3	79.2	74.6

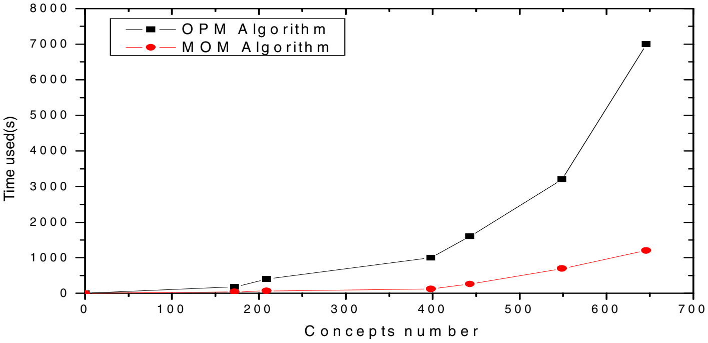


Fig. 4. Testing result comparison between the MOM and OPM

Table 2 shows the comparison between OPM and MOM. From Table 2, we can found some results of MOM are not as good as the results of OPM. But, from Fig.4, we know the time cost of MOM is much less than the cost of OPM. After analyzing the whole process of MOM, we find the reason which affects mapping accuracy. For some well designed large ontologies, the E-connection based partition approach is effective. But it is not suitable for some poor organized ontologies. A few of uncertain nodes can not be assigned to the correct module. So, in the future, we will develop new partition method to fit all kinds of ontologies.

4 Conclusion and Future Work

Large and complex ontologies are still not well supported by current ontology matching prototypes, thereby limiting the practical applicability of such systems. We propose a modularization-orient approach to decompose a large match problem into smaller ones and use a method to significantly reduce the mapping time. Our technique includes sub-steps for large ontology partitioning, finding similar modules, module matching and result combination. The experiments show that our approach is more effective in mapping the large scale ontologies than the traditional approach which directly matches the two large ontologies.

Acknowledgments

The authors would like to thank the Natural Science Funds of China (Grant No. 60374071) and National Basic Research Program of China (Grant No. 2003CB316905) for the financial support of this research.

References

1. M. Jayant, A. B. Philip, and R. Erhard. Generic Schema Matching with Cupid, in *Proceedings of the 27th International Conference on Very Large Data Bases*: Morgan Kaufmann Publishers Inc., 2001.
2. H.H.Do and E. Rahm. COMA - a system for flexible combination of schema matching approaches, in *Proceedings of VLDB* 2001, pp. 610-621.
3. D. AnHai, M. Jayant, D. Robin, D. Pedro, and H. Alon. Learning to match ontologies on the Semantic Web. The VLDB Journal, 12(4): 303-319, 2003.
4. M. Sergey, R. Erhard, and A. B. Philip. Rondo: a programming platform for generic model management, in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. San Diego, California: ACM Press, 2003.
5. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-Match: an algorithm and an implementation of semantic matching, in *Proceedings of ESWS*, 2004, pp. 61-75.
6. B. C. Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Modularizing OWL Ontologies, in *the 4th International Semantic Web Conference (ISWC-2005)*, 2005.
7. J. Hopcroft and R. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM Journal on Computing 2(4)225–231, 1973.
8. A. Maedche and S. Staab. Measuring similarity between ontologies, in *Proceedings of EKAW*, 2002.
9. Z. Wang, Y. Wang, S. Zhang, G. Shen, and T. Du. Ontology Pasing Graph-based Mapping: A Parsing Graph-based Algorithm for Ontology Mapping. Journal of Donghua University, 23(6), 2006.
10. H.H.Do, S.Melnik, and E.Rahm. Comparison of schema matching evaluations, in *Proceedings of workshop on Web and Databases*, 2002.

Finding Important Vocabulary Within Ontology

Xiang Zhang, Hongda Li, and Yuzhong Qu

Department of Computer Science and Engineering, Southeast University,
Nanjing 210096, P.R. China
{xzhang, hdli, yzqu}@seu.edu.cn

Abstract. In current Semantic Web community, some researches have been done on ranking ontologies, while very little is paid to ranking vocabularies within ontology. However, finding important vocabularies within a given ontology will bring benefits to ontology indexing, ontology understanding and even ranking vocabularies from a global view. In this paper, Vocabulary Dependency Graph (VDG) is proposed to model the dependencies among vocabularies within an ontology, and Textual Score of Vocabulary (TSV) is established based on the idea of virtual documents. And then a Double Focused PageRank algorithm is applied on VDG and TSV to rank vocabulary within ontology. Primary experiments demonstrate that our approach turns out to be useful in finding important vocabularies within ontology.

1 Introduction

An ontology presents a conceptual framework, which includes the description of concepts and relations between them. While an identical or similar set of concepts may be represented by different ontologies, it is not easy to investigate into all of the relevance ontologies totally by human effort when knowledge engineers are willing to utilize the vocabulary of others. Harith Alani terms this condition the knowledge re-use conundrum [1]. To solve the problem of reuse, ontology search engines and libraries emerge. Among them, Swoogle uses a hyperlink analysis approach for ontology ranking [2]. While researchers have depicted some major problem of ontology ranking and gave their solutions, another problem still remains unresolved. Given an ontology, some vocabularies play a crucial role when using them to define others. We refer them as "important" vocabularies in an ontology. While the "importance" of a concept has been mentioned in [3] when visualizing large-scale ontologies, the ranking scheme is still simple.

We believe that ranking vocabularies within ontology is beneficial for ranking vocabularies across ontologies such as the research by Swoogle, since finding important vocabularies in a global view can be enhanced by the local ranking of vocabularies within ontology. Besides, it is also beneficial for applications on ontology visualization or summarization, in which application users are generally only concerned with most important vocabularies described in an ontology. It is also significant for ontology indexing considering that important vocabularies tend to be retrieved more frequently among others.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 106 – 112, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Related Works

In this section, we firstly give a picture to the related works on finding important vocabularies within ontology, and then introduce a Double Focused PageRank algorithm, which is the basic ranking algorithm applied in our approach.

2.1 Ontology Ranking

Presented in [1], AKTiveRank is a ranking system closely related to our work. While the methodology of Swoogle is query-independent and the structure inside the ontology is less considered, AKTiveRank is based on the analysis of concept structure and gave a consideration on the measurement of relevence of result ontologies to multiple query keywords. The ranking of the ontology should reflect the importance of each relevant vocabulary in the ontology, which is quantified by the "Centricity" and "Density" of the vocabularies where "Centricity" represents the distance of a vocabulary to the centual level of its hierarchy, and "Density" of a vocabulary is a weighted summation of the number of its subclasses, siblings, relations and instances.

KeWei Tu described the approach of finding and presenting important classes to users in [3]. Important classes are the classes in high level of the hierarchy or having many descendants. The "importance" of a class can be quantified using a parameterisable formula, which calculates a linear combination of total "importance" of its direct subclasses and a function of its depth in the class hierarchy.

Comparing to the related ranking schemes mentioned above, our work contributes to a combined analysis of both structure and textual information when finding important vocabularies within ontology.

2.2 Double Focused PageRank

Double Focused PageRank is a variation of PageRank and Michelangelo interpreted this ranking algorithm using a unified probabilistic framework in [4]. Imagining a scene of web surfing, Double Focused PageRank defines two atomic actions for a surfer to choose when he is staying at some page: following a hyperlink from current page or jumping to another page.

In practice, the surfer will not jump or forward to other pages randomly. Considering the textual information, the surfer often prefers to jump or forward to pages with their topics relevant to the surfer's interest. In the algorithm, a text classifier is used to attach a score to each page representing its relevance to a given topic of interest. The ranking of each page is finally obtained by iteratively computing the probability distribution of the surfer staying at certain page at certain step.

The reason of choosing such algorithm for ranking vocabularies within ontology is that: Given a graph and a set of textual information attached to each node, Double Focus PageRank presents a clear and parameterisable ranking scheme considering both effect of the graph structure and textual information to the result of ranking. This feature is appropriate to determine the "importance" of vocabularies in our approach and make us different with other approaches purely using structure information.

3 Ranking Vocabularies Within Ontology

Since the nature of RDF graph is different from the graph of web pages, it is unsuitable to rank vocabularies within an ontology using link analysis directly on RDF graph: RDF graphs can't explicitly show all the dependencies between vocabularies. In an RDF graph, two vocabularies are adjacent only when they appear in a same triple. But commonly, two vocabularies may have relation via one or more medium blank nodes, which is implied by a path between the two vocabularies in RDF graph. Besides, users are mainly concerned with domain vocabularies, so blank nodes, literals and build-in vocabularies should be discarded in the ranking process.

To achieve a more reasonable vocabulary-ranking scheme, a new graph model is need, which should explicitly exhibits the dependencies between vocabularies. And similar to the interpretation of page ranking, we believe that a vocabulary is intuitively "important" if it is an authoritative node in our graph model.

3.1 RDF Sentence

Definition 1 (B-Triple). A triple in an RDF graph is called a B-Triple if it has at least a blank node as its subject or object.

Definition 2 (B-Connectedness). Two B-Triples in an RDF graph, denoted by b_s and b_t, are said to be B-Connected if and only if one of the following conditions is satisfied:

- b_s and b_t have a blank node in common;
- There exists a sequence: $b_0\,(=b_s)$, b_1, …, $b_n\,(=b_t)$ with n>1 such that b_{i-1} and b_i are B-Connected (for i =1, …n).

Definition 3 (RDF Sentence of an RDF Graph). Given an RDF graph, a triple in the RDF graph with no blank node as its subject or object is a simple RDF Sentence; a maximum subset of B-Connected B-Triples in the RDF graph is called a complex RDF sentence. Simple or complex RDF sentences are both RDF sentences (or sentence in short) of the RDF graph. And no more are called RDF sentences of the given RDF graph. The **Size** of a sentence is the number of triples in the sentence. Shown as Figure 1, two sentences can be parsed from a simple RDF graph.

Definition 4 (Subject of an RDF Sentence). The subject of an RDF sentence is the domain vocabulary playing the role as the subject in a certain triple contained by the RDF sentence supposing such domain vocabulary exists and is unique.

3.2 Vocabulary Dependency Graph

Definition 5 (Domain Vocabularies). Domain vocabularies of an RDF graph are the URI references (or URIrefs in short) that occur in the RDF graph and are not belonged to the built-ins provided by Ontology Language such as RDF or OWL.

Definition 6 (Vocabulary Dependency Graph of an RDF Graph). Vocabulary Dependency Graph (VDG), written by <V, E, W>, is an weighted directed graph such that the vertices in V are domain vocabularies in the RDF graph, and there exists an edge between two vertices if their associated domain vocabularies co-occur in at least one RDF sentence of the RDF graph. W: $E \to R^+$ and $w(i,j)$ denotes the strength of dependency between domain vocabulary *i* and *j*. The strength is formulated as a function of the size of RDF sentence, in which two vocabularies co-occur. The size can be simply deemed as the distance between the vocabularies. Dependency is directed: a vocabulary being the subject of an RDF sentence is depended on other co-occurring vocabularies with the formulated strength, and meanwhile co-occurring vocabularies also have a reversed dependency on the subject with a weaker strength.

The idea of VDG is somewhat similar to Dependency Graph proposed in [7]. However, the Dependency Graph in [7] only extracts dependencies corresponding to the subclass hierarchy and dependencies created by the domain and range restrictions in property definitions.

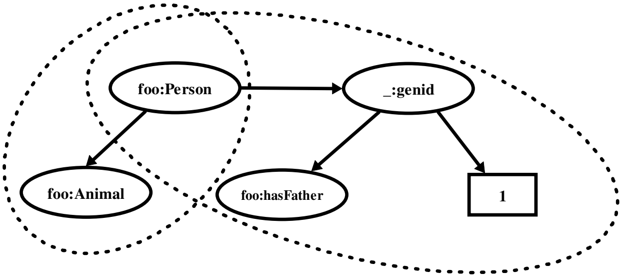


Fig. 1. A sample RDF graph and its RDF sentences

3.3 Textual Score of Vocabulary

In our model, textual score of each vocabulary is established based on the idea of virtual documents approach as described in [5] to reflect the similarity of natural language between a vocabulary and the whole ontology. A virtual document (VDoc) is a collection of weighted words containing the local descriptions and neighboring linguistic information of an URIref to reflect the intended meaning of the URIref. In an ontology, local descriptions include local name, labels, comments and other annotations of declared URIrefs. The paper mentioned above also presented the construction of VDocs and measurement of similarity between VDocs.

The original purpose to define virtual documents is for ontology mapping. In our model, each VDoc of a vocabulary is constructed taking no account of neighboring linguistic information for simplicity. A VDoc of the whole ontology is constructed by combining all the VDocs and the linguistic information of the ontology itself, including ontology comments, file name of ontology document and so on. The Textual Score of Vocabulary (TSV) is defined as the similarity between the VDoc of the vocabulary and the VDoc of the ontology as whole. Therefore, an n-dimensional vector of textual scores is then constructed, with each dimension's value indicating the relevance of each domain vocabulary to the topic of the ontology.

3.4 Ranking Process

As described in Section 2.2, Double Focused PageRank provides a ranking algorithm considering both link structure and textual content. Considering the ranking of vocabularies, a surfer might be interest with a vocabulary, and when he decides to know more about the vocabulary he will forward one of the links in VDG and take a look at the adjacent vocabulary. Heavily weighted link shows that the adjacent vocabulary is close to the original one and will be accessed with more probability. The surfer might choose to jump if he loses interest to the current vocabulary. Although he may jump arbitrarily in VDG, it is believed that he prefers the vocabulary more relevant to the topic of the ontology. The ranking of vocabularies is finally determined by the probabilistic distribution.

4 Experiments

We have performed experiments on a set of relatively small ontologies for the sake of human evaluation. In this section, we will present the experimental result of two sample ontologies: Animal ontology and Music ontology, and give our evaluation to the result.

The Animal ontology[1] is a very small ontology presenting a conceptual framework of Person (as a subclass of Animal) and relationships between persons, such as parent, spouse and friend. The RDF graph and class hierarchy of this ontology can be found at our website[2,3]. The top ten ranked vocabularies are shown in Table 1. While we treat classes and properties equally as vocabularies in the ranking process, we separate them in the final ranking list.

Another experiment is on the "Music" ontology[4]. Its class hierarchy graph and property graph can also be found at our website[5, 6]. The top ten ranked classes and properties are shown separately in Table 2. Most top ranked vocabularies are intuitively important among other vocabularies within this ontology and they are all relevant to the topic of the ontology.

Table 1. Top ten vocabularies within "Animal" ontology

	Classes Ranking		Properties Ranking	
	Local Name	**Score**	**Local Name**	**Score**
No.1	Animal	0.19360	hasAncestor	0.10708
No.2	Person	0.14556	hasParent	0.07476
No.3	Male	0.07522	hasFather	0.05199
No.4	Female	0.07216	hasMother	0.04950
No.5	Woman	0.03412	hasSpouse	0.03365
No.6	Man	0.02989	hasFriend	0.02210
No.7	HumanBeing	0.01241	hasChild	0.01652
No.8	TwoLeggedPerson	0.00676	hasMaleParent	0.01606
No.9	TwoLeggedThing	0.00334	hasFemaleParent	0.01455
No.10			biologicalMotherOf	0.01388

[1] http://www.atl.lmco.com/projects/ontology/ontologies/animals/animalsA.owl

[2] http://xobjects.seu.edu.cn/project/falcon/questionnaire/graph/animalsA_graph.htm

[3] http://xobjects.seu.edu.cn/project/falcon/questionnaire/class hierarchy/animalsA_graph.htm

[4] http://webster.cs.uga.edu/~janik/2004-Fall/8350/ontology/Music.owl

[5] http://xobjects.seu.edu.cn/project/falcon/questionnaire/class hierarchy/Music_graph.htm

[6] http://xobjects.seu.edu.cn/project/falcon/questionnaire/property%20graph/Music_graph.htm

Table 2. Top ten vocabularies within "Music" ontology

	Classes Ranking		Properties Ranking	
	Local Name	**Score**	**Local Name**	**Score**
No.1	Musical_Instrument	0.0542	has_tempo	0.0363
No.2	Musician	0.0512	plays_instrument	0.0318
No.3	Group	0.0472	consist_of_movements	0.0315
No.4	Music_piece	0.0453	belongs_to	0.0274
No.5	Movement	0.0353	owns_instrument	0.0272
No.6	String_instruments	0.0325	play_in_ensemble	0.0223
No.7	Performer Piano	0.0236	consist_of_members	0.0221
No.8	Piano	0.0221	used_instruments	0.0180
No.9	Trio	0.0191	composed_for	0.0169
No.10	Violin	0.0178	plays_piano	0.0162

5 Conclusions and Future Work

We present in this paper our novel approach to find important vocabularies within a given ontology. The idea of Double Focus PageRank is utilized for the ranking of vocabularies within ontology by considering both the structure and textual content. The structure of an ontology is characterized by a Vocabulary Dependency Graph, and Textual Score of Vocabulary is proposed to reflect the relevance of each vocabulary with the ontology. From the experimental result, our approach turns out to be useful in finding "important" vocabularies within a given ontology.

According to [6], OWL ontology can also be mapped to RDF graph to define description formulations. Because our current work is based on the RDF graph of ontology, it can be extended to reflect OWL features in the result of ranking by considering the type of edges and specifying a corresponding weighing scheme in the RDF graph when building Vocabulary Dependency Graph.

One of our future works is to consider the problem of multiple topics. Some ontologies describe conceptual frameworks on more than one topics. With current ranking scheme, important vocabularies in the light-weighted topics will be drowned by the ones in heavy-weighted topics. However, some users are willing to see the ontology separated into different partitions according to its topics and vocabularies are ranked in the range of partitions. It will be interesting to address this issue. Another future work will be ranking vocabularies across multiple ontologies by utilizing the ranking of vocabulary within each ontology.

Acknowledgements

The work is supported partially by the 973 Program of China under Grant 2003CB317004, and in part by the JSNSF under Grant BK2003001. The third author of this paper is also supported by NCET (New Century Excellent Talents in University) program under Grant NCET-04-0472. We would like to thank our colleagues for their valuable suggestions.

References

1. Alani, H., Brewster, C.: Ontology ranking based on the analysis of concept structures. In Proceedings of Third International Conference on Knowledge Capture (K-Cap), pp. 51-58, Banff, Alberta, Canada. (2005) 51-58
2. Ding, L., Pan, R., Finin, T.W., Joshi, A., Peng, Y., Kolari, P.: Finding and Ranking Knowledge on the Semantic Web. International Semantic Web Conference 2005 (2005) 156-170
3. Tu, K., Xiong, M., Zhang, L., Zhu, H., Zhang, J., Yu, Y.: Towards Imaging Large-Scale Ontologies for Quick Understanding and Analysis. International Semantic Web Conference 2005 (2005) 702-715
4. Diligenti, M., Gori, M., Maggini, M.: A Unified Probabilistic Framework for Web Page Scoring Systems. IEEE Trans. Knowl. Data Eng. 16(1) (2004) 4-16
5. Qu, Y., Hu, W., Cheng, G.: Constructing Virtual Documents for Ontology Matching. Accepted by the Fifteenth International World--Wide Web Conference. (2006)
6. Patel-Schneider, P.F., Hayes, P., Horrocks, I. (ed.): OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation 10 February 2004. Latest version is available at http://www.w3.org/TR/owl-semantics/
7. Stuckenschmidt, H., Klein, M.: Structure-Based Partitioning of Large Class Hierarchies. In Proceedings of the 3rd International Semantic Web Conference (2004) 289-303

Ontology-Based Similarity Between Text Documents on Manifold

Guihua Wen[1], Lijun Jiang[1], and Nigel R. Shadbolt[2]

[1] School of Computer Science and Engineering,
South China University of Technology, Guangzhou 510641, China
crghwen@scut.edu.cn, cssturat@sohu.com
[2] School of Electronics and Computer Science,
University of Southampton, Southampton SO17 1BJ, United Kingdom
nrs@soton.ac.uk

Abstract. This paper firstly utilizes the ontology such as WordNet to build the semantic structures of text documents, and then enhance the semantic similarity among them. Because the correlations between documents make them lie on or close to a smooth low-dimensional manifold so that documents can be well characterized by a manifold within the space of documents, we calculate the similarity between any two semantically structured documents with respect to the intrinsic global manifold structure. This idea has been validated in the conducted text categorization experiments on patent documents.

1 Introduction

With the rapid development of World Wide Web, the amount of text information is also increasing rapidly. Text document analysis such as text categorization has become the key technology in organizing and processing the large amount of text information. The task of text categorization is to assign predefined categories to text. Many existing approaches for text categorization are based on statistics and machine learning techniques [1]. The famous approaches include Naive Bayes, KNN, Linear Least Squares Fit, Neural Boosting, Support Vector Machine and so on[2]. These approaches are being improved. For example, the supervised clustering techniques have been exploited to build text categorization systems [4], while the hierarchical structure of categories is also exploitted to improve the performance further [6]. These approaches can be also combined to further improve the performance [3].

However, most approaches cannot deal with text categorization well using the similarity measures between two documents [5]. One reason is that they depend on the vector space model that assumes the items of the text document are not correlated. We solve this problem by devising similarity measures that utilize an ontology to obtain semantic structure of each document, thus comparing structures rather than terms to improve the performance of document analysis such as text categorization[13,14,15]. Another reason is that text categorization problems normally involve an extremely high dimensional space (e.g., exceed

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 113–125, 2006.
© Springer-Verlag Berlin Heidelberg 2006

30,000), but it leads to a very sparse data representation. Most approaches still directly calculate the similarity between any two documents in the high dimensional space. They do not consider that these documents may be intrinsically located on the low-dimensional manifold [11], where metric on manifold should be applied. The ideal approach should be able to crucially model the non-linear geometry of these low-dimensional manifolds [12]. Although, there is an approach that exploits the manifold to improve the Support Vector Machines for text classification [11]. This approach may be hard to have a closed-form formula to compute geodesic distances on manifolds with complex structure. It is also too complicated to implement and understand. We directly exploit the manifold learning techniques to overcome this obstacle, such as using graph theory to estimate the geodesic distance on documents manifold [8]. This approach is general, simple to understand and implement, so that it can be easily extended to many domains. Our approach is also different from those supervised nonlinear dimensionality reduction approaches such as supervised LLE [9] and supervised Isomap [10]. They utilize the class labels to improve the dimensionality reduction process, but they do not apply the properties of manifold to improve the data classification approaches directly.

2 Semantic Structure of Text Document

In text document analysis, generally text documents are treated as vectors in an n-dimensional space, where every dimension corresponds to a term (e.g.,word) and the different items are assumed to be irrelevant. Then the metrics such as the cosine of the angle between two documents can be defined. However these items are possibly interrelated. Their relationship can be recognized by the related ontology. A basis ontology defines a set of atomic concepts and situates these a concept inclusion lattice to express the domain or world knowledge, which basically is a taxonomy over concepts. Therefore this paper apply ontology such as WordNet to construct the semantic structure for each document in terms of the following two steps.

Step 1. Construct vector for each document.

We use the vector space model to represent text document. Each text document is represented as a vector using the conventional *tf-idf* (term frequency-inverse document frequency) approach to calculate the weight of item

$$w_{ij} = w(t_i, d_j) = (1 + log(tf(t_i, d_j))) \cdot \frac{N}{df(t_i)}$$

where $tf(t_i, d_j)$ is a term frequency of term t_i within document d_j, N is the total number of documents, and $df(t_i)$ is the number of documents in which term t_i appears. $w(t_i, d_j)$ is defined to be 0 when $tf(t_i, d_j) = 0$ or $df(t_i)$=0.

When all text documents are mapped, the vector matrix of text documents and the vector space of text documents spanned by $(t_1, \cdots, t_i, \cdots, t_n)$ is constructed.

Step 2. Construct semantic structures from vectors.

Each text vector consists of words. These words can be applied to automatically construct the semantic structure for document, because these words are generally semantically related. For example, if a text document contained the word *apple*, it would be related to another word *fruit* since *apple* is a kind of *fruit*. These semantically related words form the hidden but the cohesive structure of the document. Cohesion is what helps a text to hang together as a whole. Thus we use WordNet database that relates words by their meaning to construct the hidden semantic structure for any text document. The constructed structure is a sub-ontology of the WordNet ontology. This constructing process includes the following basic steps:

1. Generate list L of the most relevant terms from a vector for the given text document d
2. Let V be all nodes labeled by elements from L
3. To generate Edges set E by connecting all nodes x and y from V that are directly connected in WordNet
4. Expand (V, E) upward as following:
   - $V' = \{y|(x, y) \in E\}$
   - while $V' \neq \phi$ (1)$E' = 0$ and $B = 0$ (2)For all $x \in V'$ do if $(x, y) \in$ WordNet then $E' = E' \cup \{(x, y)\}$ and $B = B \cup \{y\}$ (3)$V = V \cup V'$; $E = E \cup E'$; $V' = B$
   - Construct a root node that connects the separated graphs together
5. Final constructed ontology for text document d is (V, E)

In order to illustrate the procedure of constructing the document semantic structure from a text document, we give an example that is a Chinese patent document consisting of identifier, title, and abstract, shown as follows:

(CH220082A) Transport apparatus for cabinet devices, especially refrigerators

The transport apparatus for a refrigerator or the like includes a frame structure which can be screwed onto the rear panel of the device and has projecting runner elements. Runner elements of this type project at all corners of the frame and lie in a common sliding plane which is parallel to the rear wall of the device when the device is to be transported horizontally. Those runner elements which are arranged near the lower edge of the rear panel of the device in addition extend beyond the aforementioned lower edge if the device is to be transported upright. These measures make it possible for one person to transport, without effort, even relatively large and heavy refrigerators or the like in a horizontal or upright position in a gentle manner.

Because constructed semantic structure of the document is very complicated if all words extracted from patent text document are utilized, for simplicity, we choose several most important noun words as example. These noun words are **transport, apparatus, cabinet, refrigerator, frame, structure, and panel**. According to constructing algorithm, The constructed graph for this

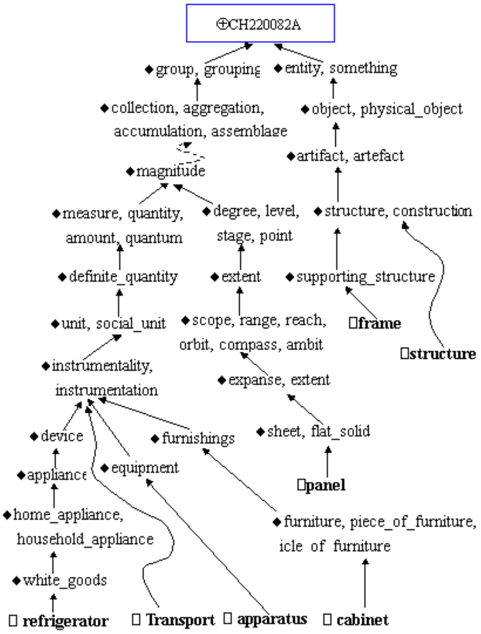


Fig. 1. Semantic structure constructed from text

patent text document is shown as Figure 1, which is named with the patent identifier because it is unique.

3 Measures on Manifold

When all text documents are semantically structured using the shared ontology such as WordNet, the semantic similarity between documents can be established on their hierarchical semantic structures.

3.1 Similarity Between Concepts

Based on the hierarchical structure of the shared ontology such as WordNet, many approaches calculate the similarity between two concepts using path distance between concepts in the hierarchical structure underlying the shared ontology. For example, the following defined measure is based on that concepts at upper layers of the hierarchy have more general semantics and less similarity between them, while concepts at lower layers have more concrete semantics and stronger similarity [13], where h_x is the height of the node x in the hierarchy, h_y is the height of the node y in the hierarchy, and h_{xy} is the height of the node of greatest depth that is an ancestor of both x and y in the hierarchy.

$$sim_H(x, y) = \frac{2 \times h_{xy}}{h_x + h_y}$$

This measure does not consider the contribution of path length to the similarity. A better similarity measure is posited to consider simultaenously the shortest path length l as well as the depth of the subsumer. It scales down similarity for subsuming concepts at upper layers and to scale up similarity for subsuming concepts at lower layers [17].

$$sim_H(x, y) = e^{\alpha l} \cdot \frac{e^{\beta h_{xy}} - e^{-\beta h_{xy}}}{e^{\beta h_{xy}} + e^{-\beta h_{xy}}}$$

This measure does not scale the similarity. Based on the relative depth of the least common super-concept between these two concepts to respective depths, we combine above two measures to define a new measure

$$sim_H(x, y) = e^{\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}$$

where $h = 2h_{xy}/(h_x + h_y)$, $\alpha, \beta > 0$ are parameters scaling the contribution of shortest path length and relative depth respectively.

3.2 Similarity Between Documents

Based on the similarity measure on concepts, the vector-based cosine similarity measure and Euclidean distance can be adapted to define the similarity of two documents, for instance, by exploiting their hierarchical structure, where the assumption that the different components of the vector are perpendicular to each other are dropped [13]. One advantage of this approach is that we can weight the components of the vectors, by using schemes such as *tf-idf*. For example, semantic structure of the document can be transformed into their corresponding preorder or postorder traversal sequences, and then the corresponding vectors can be constructed. From them, the similarity between documents can be computed, which generally can lead to the better performance for applications. Suppose that two documents $\mathbf{x}$ and $\mathbf{y}$ can be represented vectors respectively as follows, where x_i is a concept of the document and a_i is the weight of x_i.

$$\mathbf{x} = \sum_{i=1}^{m} a_i x_i$$

$$\mathbf{y} = \sum_{j=1}^{n} b_j x_j$$

Now we redefine exactly the dot product of two documents based on their semantic structures

$$\mathbf{x} \otimes \mathbf{y} = \sum_{i=1}^{m}\sum_{j=1}^{n} a_i b_j x_i \otimes y_j = \sum_{i=1}^{m}\sum_{j=1}^{n} a_i b_j sim_H(x_i, y_j)$$

This equation is identical to the standard vector space model, except that $x_i \cdot y_j \neq 0$ when $x_i \neq y_j$. From this equation, we can redefine the cosine similarity and structured Euclidean distance between documents respectively as follows:

$$sim(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}\otimes\mathbf{y}}{\sqrt{\mathbf{x}\otimes\mathbf{x}}\sqrt{\mathbf{y}\otimes\mathbf{y}}}$$

$$d_T(\mathbf{x},\mathbf{y}) = \sqrt{\mathbf{x}\otimes\mathbf{x} - 2\mathbf{x}\otimes\mathbf{y} + \mathbf{y}\otimes\mathbf{y}}$$

Sometimes, the similarity of documents can be defined by distance measure as long as without changing the logic relationships among documents, such as

$$sim(x,y) = \frac{1}{1 + d_T(x,y)}$$

Where and what measure are applied depends on the context. However, despite this function is monotonic, it may change the logic structure of a set of concepts or documents when it is applied in reasoning.

3.3 Measuring Similarity on Manifold

In many real-world problems, the vector representation of most documents are high-dimensional points in space, but they typically have a much more compact description. Coherent structure in the world leads to strong correlations between ontologies, generating observations that lie on or close to a smooth low-dimensional manifold. To compare and classify them-in effect, to reason about the world-depends crucially on modeling the nonlinear geometry of these low-dimensional manifolds [8,12]. Our goal is to calculate their similarity with respect to the intrinsic global manifold structure collectively revealed by a large number of documents. Thus any document and its concepts may be well characterized by a manifold within the space of documents, on which geodesic distance on the manifold is utilized instead of Euclidean distance.

On documents manifold M, the length of a smooth curve $\Gamma : [0,1] \rightarrow M$ is defined as

$$l(\Gamma) = \int_0^1 |\frac{d}{dt}\Gamma(t)|dt$$

The geodesic distance between points $x, y \in M$ is the length of the shortest (piecewise) smooth curve between the two points

$$d_T^g(x,y) = \inf_{\Gamma}\{l(\Gamma) : \Gamma(0) = x, \Gamma(1) = y\}$$

In such case, direct distance such as defined above is meaningful between the nearby points only, geodesic distance is more suitable on manifold. The difference between them can be illustrated in Figure 2 where the spiral is embedded in a two-dimensional space.

It has been shown that for sufficiently dense sampling, geodesic distance can be estimated approximately from an direct distance graph connecting all local neighborhoods of points [8]. The nearest approach connects each point to its k nearest neighbors are most popular approaches currently used for constructing

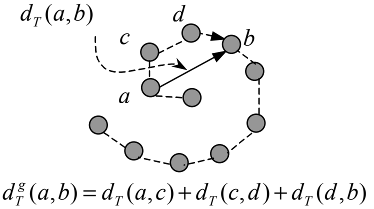


Fig. 2. Difference between Euclidean and geodesic

a neighborhood graph. We utilize this approach to estimate approximately the geodesic distance between points, which primarily consists of two steps:

Step 1. Construct the weighted graph $G = (V, E)$ for a set of documents, by connecting each document to all its k-nearest neighbors, where $(x, y) \in E$ if document x is a member of k neighbors of x by using distance $d_T(x, y)$ to make decision.

Step 2. The graph G is then used to approximate the geodesic distance $d_g(x, y)$ between any two documents as the shortest path through graph that connects them.

1. Initialize
$$d_T^g(x,y) = \begin{cases} d_T(x,y) : (x,y) \in E \\ \infty : (x,y) \notin E \end{cases}$$
2. Iterate the following formula to generate the geodesic distance between two documents using all $z \in V$
$$d_T^g(x,y) = \min_z \{d_T^g(x,y), d_T^g(x,z) + d_T^g(z,y)\}$$

This algorithm is motivated by the fact that locally a smooth manifold is well approximated by a linear hyper-plane and, so, geodesic distances between neighboring documents are close to their direct distances. For faraway documents, the geodesic distance is estimated by summing the sequence of such local approximations over the shortest path through the graph. In order to illustrate the influence of geodesic distance and Euclidean distance on determining the neighborhood, we give an example. in Figure 3, the categories of *6*-nearest neighbors of x using Euclidean distance, encompassed by big circle, is

$$N_e(x) = \{square, square, square, circle, circle, circle\}$$

This neighborhood is not consistent with our intuition, because points with square categories are not neighbors of x on manifold. By contrast, if geodesic distance is utilized to determine the neighbors for x, the result is

$$N_g(x) = \{circle, circle, circle, circle, circle, circle\}$$

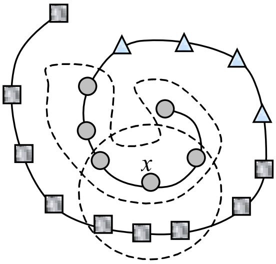


Fig. 3. Different neighborhoods using Euclidean distance and geodesic distance

This is as we expected, which leads to the correct result. From the geodesic distance, we can get the similarity measure on manifold as follows

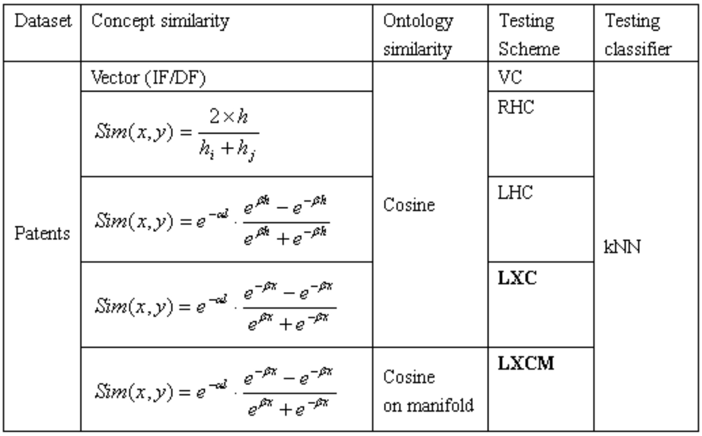
$$sim_g(x, y) = \frac{1}{1 + d_T^g(x, y)}$$

Now we can apply the similarity measure on manifold to measure the documents in applications. It can be observed that the measures on manifold depend on the direct measures d_T between documents. These direct measures not only include those defined above, but also refer to any other existing ones that have general distance semantics. In this sense, measures on manifold do not contradict with the existing meausres. The ideas behind them can be also applied to extend the existing measures to manifold.

4 Experimental Results

In order to validate the proposed similarity measures, we apply them to text categorization on patent text documents. This dataset consists of **355** samples with **6** classes. We choose this dataset for experiments with two reasons. The first reason is that the majority of ontology learning methods developed so far exploit textual sources [18], we are also investigating to learn ontologies from patent documents and then exploit them for patent analysis[12]. We hope these measures can directly serve this purpose. The second reason is that a huge amount of patent documents are available that they have been classified according to IPC (international patent classification) classes by official patent officers so that they are suitable for text categorization. The whole dataset is randomly split using "ModApte" into two parts: 70% documents for training and the other 30% for testing [5]. This kind of random division are performed ten times for experiments.

In experiments, five schemes illustrated as Figure 4 will be compared in terms of classification accuracy of documents, where scheme 4 and scheme 5 are our proposed. The k-nearest-neighbor (kNN) is selected as a baseline classifier, as it is extremely simple to implement and often results in very good classification performances in many practical applications [1].



Dataset	Concept similarity	Ontology similarity	Testing Scheme	Testing classifier
Patents	Vector (IF/DF)	Cosine	VC	kNN
	$Sim(x,y) = \frac{2 \times h}{h_i + h_j}$		RHC	
	$Sim(x,y) = e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}$		LHC	
	$Sim(x,y) = e^{-\alpha l} \cdot \frac{e^{-\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}}$		**LXC**	
	$Sim(x,y) = e^{-\alpha l} \cdot \frac{e^{-\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}}$	Cosine on manifold	**LXCM**	

Fig. 4. Experimental design

The whole experiment process includes three steps. We construct a vector for each document, and then build its semantic structure from this vector according to WordNet 1.6, where we only exploit the noun portion of WordNet that is the most developed part of the network and within it the subsumption hierarchy (hypemymy/hyponym) makes up over 80% of the links. In the third step, we construct the documents manifold. Then kNN is applied to classify these documents with attempts to compare five testing schemes.

Experiment 1. In order to investigate whether the proposed similarity measure between concepts is superior to the other two measures in terms of classification error, we conducted experiments using structured cosine similarity measure, where all parameters are tested to choose the least error. The scope of parameters are set up as k=6-20 for KNN, α and β taking 10 groups random parameters for LHC and LXC. It can be observed from Figure 5 that (1)RHC, LHC and LXC all perform better than VC ,which illustrates the WordNet ontology can be applied to improve the classification accuracy of text documents. (2)LHC and LXC perform further better than RHC. This indicates semantic similarity between concepts should be determined by combining multiple information sources nonlinearly. (3)In most cases LXC also further performs better than LHC, which illustrates semantic similarity between concepts in hierarchy should take relative height instead of absolute height. It stands for the proposed similarity measures on concepts.

Experiment 2. In order to check whether our method is effective when it is extended to manifold, we do experiments to compare among VC, LXC, and LXCM in terms of classification error. It can be observed from Figure 6 that our proposed measure on manifold is further improved, namely VC< LXC<LXCM, where LXC and LXCM are all based on ontology but LXCM are also on manifold. It illustrates that space constructed from documents is the curved manifold where geodesic distance should be applied.

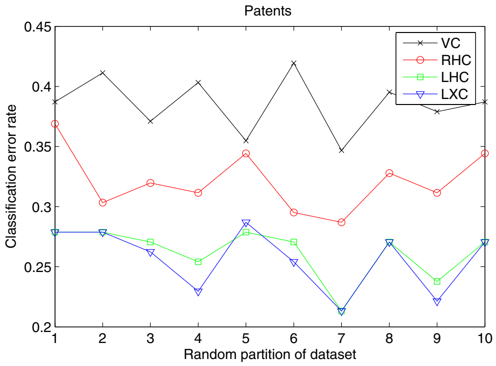


Fig. 5. Comparison in Euclidean space

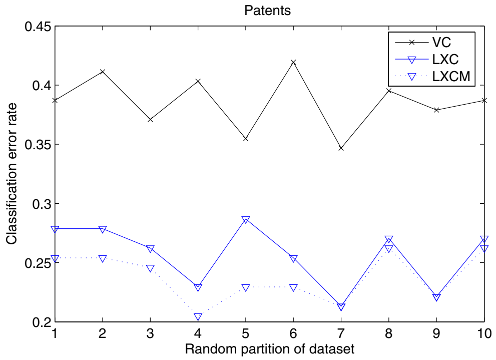


Fig. 6. The proposed measure in Euclidean and Manifold

Experiment 3. In order to prove the generality that measures on documents should be extended to manifold, we conducted experiments by extending the RHC, LHC and LXC to the manifold. It is discovered from Figure 7 that (1)While in most cases RHCM performs superior to RHC, both LHCM and LXCM consistently do better than LHC and LXC do in all cases. This illustrates that performance of approach on manifold depends on the measures used to estimate the geodesic distance. (2)On some dataset, LXC is superior to LHC in original space, but on manifold LXCM performs worse than LHCM does. This illustrates that good measure in original flat space is not surely also good on manifold. In practices, we need to directly evaluate them on manifold. (3)However, accuracy of all measures on manifold is improved with different degree than that in original space. This illustrate that measuring similarity between documents on manifold is general, not limited a specific measure or approach.

Discussions. All experimental results consistently stand for our idea that measures on structured documents should be established on manifold. In experiments, despite only one kind of data is used, this data comes from the real

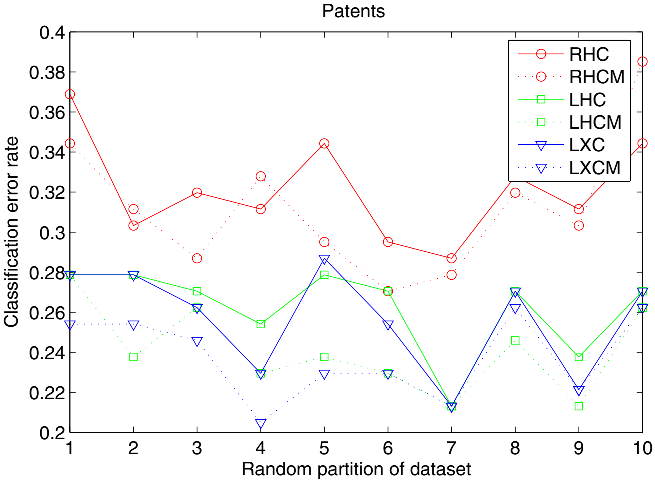


Fig. 7. Comparison in manifold

application and more it is coped technically to generate more random dataset. These strategies improve the reliability of experimental results and support for the usability of our proposed approaches. In experiments, we discover that classification accuracy depends on structure of documents that is constructed from words in text document. Currently only the front 500 words of 3115 words is used to construct the semantic structure of documents, but does it lead to the large improvement of the accuracy. We argue that if more words are utilized, the performance can be further improved. However, this is at the cost of time. To solve this problem, some key words to semantic structures of documents should be chosen by using machine learning techniques such as feature selection, instead of increasing the number of words. On manifold, additional time is needed to estimate the geodesic distance. This can be overcome technically in practical applications. For example, for text categorization with KNN classifier, the geodesic distance on training set can be calculated in advance, the geodesic distance of the new coming text document to the elements in training set can be estimated approximately with additional little time when classifying. Despite the approach is evaluated on automatically constructed semantic structures of documents, it also scales to manually made ontologies, because our measures are based on general formal ontology model independent of text document structure.

5 Conclusions

This paper makes contributions in the following aspects. It presents a new similarity measure between concepts that emphasize the nonlinear integration of shortest path and relative depth. Subsequently structured Euclidean distance between documents is also proposed. More importantly, it proposes an important idea and approach to define similarity measures between documents on manifold, thus solid mathematics can be exploited to study documents. These contributions are validated by the conducted experiments. Therefore we argue that with our investigation we have created a methodological baseline and collected some empirical experiences. In the future we will apply this approach to investigate the similarity between more general ontologies.

Acknowledgement

This work is supported by Natural Science Foundation of China(60003019), Hubei Science and Technology Project (2005AA101C17), and UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01.

References

1. Sebastiani, F: Machine Learning in Automated Text Categorization. ACM Computing Surveys **34** (2002) 1-47
2. J.He and A.H.Tan and C.L.Tan, On Machine Learning Methods for Chinese Document Categorization, *Applied Intelligence*, vol.18, 2003, pp 613- 617
3. D.A. Bell and J.W. Guan and Y.Bi, On Combining Classifier Mass Functions for Text Categorization, *IEEE Transactions on Knowledge and Data Engineering*, vol.17, 2005, 1307-1319.
4. C.C.Aggarwal and S.C. Gates and P.S. Yu, On Using Partial Supervision for Text Categorization, *IEEE Transactions on Knowledge and Data Engineering*, vol.16,2004, pp 245 - 255
5. W.Lam and Y.Q.Han, Automatic textual document categorization based on generalized instance sets and a matamodel, *IEEE Transactions on Pattern Analysis and Machine Intelligence*,vol.25, 2003, pp 628-633
6. Sun,A. and Lim,E.P. and Ng,W.K. and Srivastava,A.,Blocking Reduction Strategies in Hierarchical Text Classification, *IEEE Transactions on Knowledge and Data Engineering*, vol.16,2004, pp 1305-1308
7. S.T. Roweis and L.K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, vol.290, 2000, pp 2323–2326.
8. J.B.Tenenbaum and V.de Silva and J.C.Langford: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science **290** (2000) 2319-2323
9. de Ridder D, Kouropteva O, Okun O, et al., Supervised locally linear embedding, *Lecture Notes in Artificial Intelligence*,vol.2714, 2003, pp 333-341.
10. X.Geng and D.C.Zhan and Z.H.Zhou, Supervised Nonlinear Dimensionality Reduction for Visualization and Classification, *IEEE Transactions on Systems, Man and Cybernetics*, vol.35, 2005, pp 1098–1107.
11. D.Zhang and X.Chen and W.Lee, "Categorization and supervised machine learning: Text classification with kernels on the multinomial manifold", *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR '05*, Brazil, 2005, pp 266 - 273
12. Wen, G.H.: Rotating dynamics for computational creativity. BeiJing:National Defence Industry Press **book** (2005)
13. Ganesan, P., Molina, H.G., Widom J.: Exploiting hierarchical domain structure to compute similarity. ACM Transactions on Information Systems **21** (2003) 64-93

14. Yuan S.T., Sun J.: Ontology-Based Structured Cosine Similarity in Document Summarization: With Applications to Mobile Audio-Based Knowledge Management. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics **35** (2005) 1028 - 1040
15. Vladimir Oleshchuk, Asle Pedersen: Ontology Based Semantic Similarity Comparison of Documents, 14th International Workshop on Database and Expert Systems Applications, (2003)735,
16. Rodriguez, M.A., Egenhofer, M.J.: Determining semantic similarity among entity classes from different ontologies. IEEE Transactions on Knowledge and Data Engineering **15** (2003) 442-456
17. Li, Y., Bandar, Z.A., Mclean, D : An approach for measuring semantic similarity between words using multiple information sources. IEEE Transactions on Knowledge and Data Engineering **15** (2003) 871 - 882
18. Navigli, R.; Velardi, P.; Gangemi, A.: Ontology learning and its application to automated terminology translation. IEEE Intelligent Systems **18** (2003) 22 - 31

A Formalism of XML Restructuring Operations*

Jixue Liu[1], Ho-Hyun Park[2], Millist Vincent[1], and Chengfei Liu[3]

[1] School of Computer and Information Science, University of South Australia
{jixue.liu, millist.vincent}@unisa.edu.au
[2] School of Electrical and Electronic Engineering, Chung-Ang University
hohyun@cau.ac.kr
[3] Faculty of ICT, Swinburne University of Technology
cliu@swin.edu.au

Abstract. We present a set of primitive restructuring operators that, when combined, are sufficiently powerful to convert an XML document under a source schema into an XML document under an arbitrary target schema. We initially define the operators at the schema level, and then show how each operator induces a corresponding transformation on any XML document under the schema. Finally, we note that our operators can be implemented in a high level language such as XQuery, and thus our approach can be used as the basis for automating the conversion of one XML document to another XML document.

1 Introduction

XML has emerged as a standard for data representation and interchange on the Internet and much data has been made available in XML format. When the data from multiple sources is integrated into a global repository, source data needs to be restructured to agree with the structure of the integrated data. In order to do this, powerful restructuring operators are needed because the flexibility of XML means that the structure of the source data may vary significantly from the structure of the integrated data. In this paper we present a set of such restructuring operators. We choose DTD's (Document Type Definitions) as the schema specification language, rather that a more complex language such as XML Schema, since DTDs can be formalized easily as a context free grammar.

Restructuring operators have been proposed in some work in data integration [2,10] and have similar counterparts in XML algebras such as [3,5,7] and query languages such as XSLT and XQuery. However, as we now outline, these previous studies of restructuring operators have neither been systematic nor formal.

A number of transformation operators have been proposed for XML in the literature. In [8], operators are defined to add or delete subtrees, to add or remove intermediate nodes between a node and its descendant nodes, to replace a sub tree with a sequence of sub trees, and to split a conjunction element into an alternation (disjunction). The work [9] defines, among other operators,

* This research was supported by Australian Research Council's Discovery Project DP0559288.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 126–132, 2006.
© Springer-Verlag Berlin Heidelberg 2006

an extend operator as a surrogate so that when a parent-child relationship is reversed, no information is lost. In [2], a product operator is defined which is similar to the unnest operator in the nested relational model. A detailed analysis of the differences of between these operators in the literature and our operators is given in Section 3, but we make the general point that most of this previous work is based on examples rather than formalism, and have not considered the full DTD syntax which includes multiplicity indicators, multi-layered sub-structures, and alternation.

We make the following contributions in the paper. After presenting the preliminary definitions of the XML DTD model, the XML document model, and the conformation of a document to a DTD, the paper proposes a set of operators for restructuring both DTDs and the corresponding XML documents. Because of the syntax differences between XML DTDs and XML documents, each operator has two parts. The first part defines the processes for DTD restructuring while the second defines the processes for document restructuring. The operators are defined with the consideration of full syntax of XML DTD including multiplicity indicators, multi-layered sub-structures, and alternations. These operators enable a DTD to be restructured to any other DTD and the conforming documents to be transformed accordingly.

We formalize the definitions by giving exact semantics of the operators with regard to both DTD transformation and document transformation. This makes the semantics of the operators much more precise than those definitions appearing in the literature where the semantics of operators are given through examples. This is also one of the important differences of this paper from others.

The formalism of the paper standardizes XML restructuring operations. With this standardization, the operators can be implemented as stored procedures/ queries so that when they are needed, these procedures can be called. In this way, users are freed from composing complex queries. Our implementation show that implementing these operators is non trivial work and in many cases is challenging.

2 Preliminary Definitions

In this section we give preliminary definitions. We first present XML DTDs defined in the XML recommendation with the restrictions of no recursion and no duplicated element names in an element definition. After the definition of DTDs, we define XML trees and the conformation of XML trees to DTDs.

In the rest of this paper, given a sequence $x_1, x_2, \cdots, x_m$, we use $m \geq 0$ (note that 0 is less than the starting subscript) to denote that the sequence can be empty.

Definition 2.1 (XML DTD). A XML Document Type Definition **DTD** is defined to be $D = \langle\!\langle\ EN, G, \beta, root\ \rangle\!\rangle$ where:

(a) EN is a finite set of element names;
(b) the set of type descriptions G is defined by $g \in G$ if
$g = Str$ where Str is a symbol denoting #PCDATA (text);

$g = e$ where $e \in EN$;
$g = \epsilon$, indicating the EMPTY type;
$g = g_1, g_1$ or $g_1|g_1$ or $g_2^c \wedge (g_2 = e \wedge e \in EN$ or $g_2 = [g, \cdots, g]$ or $g_2 = [g|\cdots|g])$ where $'|'$ and $','$ are disjunction and conjunction respectively, $c \in \{?, 1, +, *\}$ is the cardinality denoted by the function $c(g_2)$, and [ ] is the type layer, $g_1 \neq Str$, $g_2 \neq Str$, $g_1 = g$ is recursively defined. g_1 and g_2 are called components and denoted by $g_1, g_2 \in g$; all element names in g are distinct;

(c) β is a function from EN to G as $\beta(e) = [g]^c$ defining the type of e;
(d) *root* is the root of the DTD and is not in EN. □

Following the definition, if $\beta(e) = [g]^c \wedge g_1, g_2 \in g$, then $par(g_1) = \beta(e)$ and $g_1 \cap g_2$ is the set of element names in both g_1 and g_2.

Example 2.1. *An example DTD is* $D_a = \langle\!\langle\ EN, G, \beta, root\ \rangle\!\rangle$ *where*
$EN = \{root, auth, work, book, article, title, name, affi, publ, conf, loc, year\}$,
$\beta(root) = [auth]^*$,
$\beta(auth) = [name, affi^*, work^*]$
$\beta(work) = [book|article]$
$\beta(book) = [title, publ, year]$,
$\beta(article) = [title, conf, loc^?, year]$,
$\beta(name) = Str$,
$\beta(affi) = \beta(publ) = Str$,
$\beta(year) = \beta(conf) = \beta(loc) = Str$.

Definition 2.2 (XML tree). Let EN be a finite set of element names, V a finite set of node identifiers, VAL an infinite set of text strings, $\perp$ a special value. An **XML tree** T is defined to be $T = (v : e : val, T_1, T_2, \cdots, T_f)$ where $v \in V$, $e \in EN$, ($val \in VAL$ or $val = \perp$), the triple $v : e : val$ is called a node, and

(i) if $val = \perp$, then $f \geq 0$, and $T_1, T_2, \cdots, T_f$ are recursively defined trees denoted by $Ch(v)$. The triple $v{:}e{:}val$ is often simplified to "␣$:e$" when the context is clear;
(ii) if $val \in VAL$, then $f = 0$. □

In the notation, the node identifiers are added to enable unique references. A layer of brackets in the notation corresponds to a level in the tree.

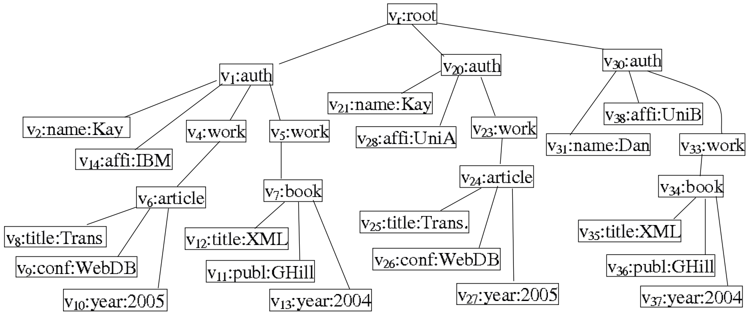


Fig. 1. An XML tree

Example 2.2. *Following is an XML tree in our notation which is also graphically represented in Fig. 1.*

$T = (v_r{:}root, (v_1{:}auth, (v_2{:}name{:}Kay), (v_{14}{:}affi{:}IBM), T_4, T_5), T_{20}, T_{30})$
$T_4 = (v_4{:}work, (v_6{:}article, (v_8{:}title{:}Trans.), (v_9{:}conf{:}WebDB), (v_{10}{:}year{:}2005)))$
$T_5 = (v_5{:}work, (v_7{:}book, (v_{12}{:}title{:}XML), (v_{11}{:}publ{:}GHill), (v_{13}{:}year{:}2004)))$
$T_{20} = (v_{20}{:}auth, (v_{21}{:}name{:}Kurz), (v_{28}{:}affi{:}UniA), (v_{23}{:}work,$
$(v_{24}{:}article, (v_{25}{:}title{:}Trans.), (v_{26}{:}conf{:}WebDB), (v_{27}{:}year{:}2005))))$
$T_{30} = (v_{30}{:}auth, (v_{31}{:}name{:}Dan), (v_{38}{:}affi{:}UniB), (v_{33}{:}work,$
$(v_{34}{:}book, (v_{35}{:}title{:}XML), (v_{36}{:}publ{:}GHill), (v_{37}{:}year{:}2004))))$.

Definition 2.3 (hedge). A hedge H is a sequence of trees $T_1, T_2, \cdots, T_n$. □

A hedge groups the child trees of a node so that the cardinality constraints of a type in a DTD can be tested.

Definition 2.4 (conformation). A hedge H conforms to a type $g \in G$, denoted by $H \Subset g$, if all of the followings are true.

(1) if $g = e$ and $\beta(e) = Str$, $H = T$ and $T = (v{:}e{:}txt)$;
(2) if $g = e$ and $\beta(e) = g_1$, $H = T$ and $T = (v{:}e{:}\bot, H')$ and $H' \Subset g_1$;
(3) if $g = \epsilon$ or $g = Str$, $H = \phi$ (empty);
(4) if $g = g_1, g_2$, $H = H_1, H_2$ and $H_1 \Subset g_1$ and $H_2 \Subset g_2$;
(5) if $g = g_1 | g_2$, $H = H_0$ and $H_0 \Subset g_1$ or $H_0 \Subset g_2$;
(6) if $g = g_1^c \wedge g_1 = e$, $H = H_1, \cdots, H_f$ and $\forall\, i = 1, \cdots, f (H_i \Subset e)$ and f satisfies c;
(7) if $g = g_1^c \wedge g_1 = [g]$, $H = H_1, \cdots, H_f$ and $\forall\, i = 1, \cdots, f (H_i \Subset g)$ and f satisfies c.

Given a DTD $D = \langle\!\langle\, EN, G, \beta, root \,\rangle\!\rangle$ and XML tree T, T conforms to D if $T = (v_r{:}root{:}\bot, H)$ and $H \Subset \beta(root)$. □

The XML tree in Example 2.2 conforms to the DTD in Example 2.1.

Definition 2.5 (equivalence).

(i) Two trees T_a and T_b are equivalent, denoted by $T_a = T_b$, if
(1) $T_a = (v_1{:}e{:}txt)$ and $T_b = (v_2{:}e{:}txt)$ or
(2) $T_a = (v_1{:}e{:}\bot, T_1, \cdots, T_m)$ and $T_b = (v_2{:}e{:}\bot, T_1', \cdots, T_n')$ and $m = n$ and for $i = 1, \cdots, m (T_i = T_i')$
(ii) Two hedges $H_x = T_1, \cdots, T_m$ and $H_y = T_1', \cdots, T_n'$ are equivalent, denoted by $H_a = H_b$, if $m = n$ and for $i = 1, \cdots, m (T_i = T_i')$.

3 DTD and Document Restructuring Operations

In this section, we propose a set of XML restructuring operators. In defining these operators, we consider the full syntax of XML DTDs and XML documents. We put special effort to handle multiplicities in the definitions so that the constraining power of multiplicities during restructuring can be realized. In addition, we also considered the handling of alternations, which to the best of knowledge no previous work has ever done so. For each restructuring operator, we define two parts to deal with DTDs and documents separately due to the fact that DTDs and documents have different syntaxes and different models. With the operators, one can transform a DTD and its conforming documents to any other DTD and the corresponding conforming documents.

We define the intervals of the multiplicities $'?'$, $'1'$, $'+'$, $'*'$ to be $[0, 1]$, $[1, 1]$, $[1, n]$, $[0, n]$ respectively. Let c_1 and c_2 be two multiplicities. Then the union $c_1 + c_2$ is defined by the union of their intervals and the intersection $c_1 \cap c_2$ by the intersection of their intervals. As a result, '+'+'?'='*', '1'+'?'='?', '+'∩'?'='1', and '1'∩'?'='1'. The difference $c_1 - c_2$ means the interval of c_1 taking away that of c_2 and union that of '1'. If c is a multiplicity, $c \geq$'1' mean c is either '1' or '+'.

We now list the operators defined in this paper in Table 3.1 where g means a list of sub types, e means an element name, $H_1^k = H(g_k)_1$ denotes the i-th hedge of g_k without considering $c(g_k)$ or $c(g_1) = 1$, and H_i^{*k} denotes the i-th hedge of $g_k^{c_k}$ where the '*' in the superscript indicates the cardinality. Note that $H_1^{*1} = H_{1_1}^1, \cdots, H_{1_{f_1}}^1$. Detailed definitions of the operators can be found in [6]. We use a few examples to show the use and the meaning of the operators. In the example, the placeholder $\sqcup$ indicates a distinct node identifier.

Example: This example shows the use of the *min* operator. Let $\beta(e) = [B, [C^*, D]^+, E]^*$ and $T_i = (\sqcup : e, (\sqcup : B), (v_1 : C), (v_2 : C), (v_3 : D), (v_4 : C), (v_5 : D), (\sqcup : E), (\sqcup : B), (v_7 : C), (v_8 : D), (v_9 : C), (v_{10} : D), (\sqcup : E))$. $min([C^*, D]^+) \rightarrow \beta(e) = [B, [C^*, D^+], E]^*$ and $min(T_i) \rightarrow T_o = (\sqcup : e, (\sqcup : B), (v_1 : C), (v_2 : C), (v_4 : C), (v_3 : D), (v_5 : D), (\sqcup : E), (\sqcup : B), (v_7 : C), (v_9 : C), (v_8 : D), (v_{10} : D), (\sqcup : E))$.

Example: This example shows the use of the *mout* operator. Let $\beta(e) = [B, [C^*, D^+]^?, E]^*$ and $T_i = (\sqcup : e, (\sqcup : B), (v_1 : C), (v_2 : C), (v_3 : D), (\sqcup : E), (\sqcup : B), (v_7 : C), (v_8 : D), (v_9 : D), (\sqcup : E))$. $mout([C^*, D^+]^?) \rightarrow \beta(e) = [B, [C^?, D]^*, E]^*$ and $mout(T_i) \rightarrow T_o = (\sqcup : e, (\sqcup : B), (v_1 : C), (v_3 : D), (\sqcup : E), (\sqcup : B), (v_7 : C), (v_8 : D), (\sqcup : E))$. Note the loss of nodes v_2 and v_9.

Example: This example shows the use of the *nest* operator. Let $\beta(t) = [B, [C^*, D]]^*$, $\beta(B) = \beta(C) = \beta(D) = Str$, and $T_i = (\sqcup : t\ (v_1 : B\ (\sqcup : Str\ 1)), (v_2 : C\ (\sqcup : Str\ 2)), (v_3 : D\ (\sqcup : Str\ 3)), (v_4 : B\ (\sqcup : Str\ 1)), (v_5 : C\ (\sqcup : Str\ 2)), (v_6 : C\ (\sqcup : Str\ 2)), (v_7 : D\ (\sqcup : Str\ 3)), (v_8 : B\ (\sqcup : Str\ 2)), (v_9 : C\ (\sqcup : Str\ 2)), (v_{10} : D\ (\sqcup : Str\ 3)))$. $nest([C^*, D]) \rightarrow \beta(t) = [B, [C^*, D]^+]^*$ and $T_o = (\sqcup : t\ (v_1 : B\ (\sqcup : Str\ 1)), (v_2 : C\ (\sqcup : Str\ 2)), (v_3 : D\ (\sqcup : Str\ 3)), (v_5 : C\ (\sqcup : Str\ 2)), (v_6 : C\ (\sqcup : Str\ 2)), (v_7 : D\ (\sqcup : Str\ 3)), (v_8 : B\ (\sqcup : Str\ 2)), (v_9 : C\ (\sqcup : Str\ 2)), (v_{10} : D\ (\sqcup : Str\ 3)))$. Note the loss of v_4.

This section formalized a set of operators for XML data restructuring. These operators are expected to give sufficient power to transform a DTD to any other DTD. We note that the operators of selection, insertion, deletion, and join are not included here. The main reason for their exclusion is that their functionality is more querying and updating than restructuring. At the same time, the definition of the join operator can be complex, which we leave for future work, because the operators may need to consider keys and recursion.

Table 1. The transformation operators

op.	**DTD**	doc
min	$[g_1^{c_1}, \cdots, g_n^{c_n}]^c \rightarrow$ $[g_1^{c_1+c}, \cdots, g_n^{c_n+c}]^1$	$H_1^{*1}, \cdots, H_1^{*n}, \ \cdots, \ H_m^{*1}, \cdots, H_m^{*n} \rightarrow$ $H_1^{*1}, \cdots, H_m^{*1}, \ \cdots, \ H_1^{*n}, \cdots, H_m^{*n}$
$mout$	$[g_1^{c_1}, \cdots, g_n^{c_n}]^c \rightarrow$ $[g_1^{c_1-c_c}, \cdots, g_n^{c_n-c_c}]^{c+c_c};$ $c_c = minc_1, \cdots, c_n$	$H_{11}^1, \cdots, H_{1d_{11}}^1, \cdots, H_{11}^n, \cdots, H_{1d_{1n}}^n, \ \cdots,$ $H_{m1}^1, \cdots, H_{md_{m1}}^1, \cdots, H_{m1}^n, \cdots, H_{md_{mn}}^n \rightarrow$ $H_{11}^1, \cdots, H_{11}^n, \cdots, H_{1w_1}^1, \cdots, H_{1w_1}^n, \cdots,$ $H_{m1}^1, \cdots, H_{m1}^n, \cdots, H_{mw_m}^1, \cdots, H_{mw_m}^n;$ $w_i = min(d_{i1}, \cdots, d_{in})(i = 1, \cdots, m)$
$rename$	change element e to e_1	$(v_1{:}e, H_0) \rightarrow (v_1{:}e_1, H_0)$
$shift$	$g_i, g_j \rightarrow g_j, g_i$	$H^i, H^j \rightarrow H^j, H^i$
$group$	$g_1, \cdots, g_n \rightarrow [g_1, \cdots, g_n]^1$	hedge unchanged
$ungroup$	$[g_1, \cdots, g_n]^1 \rightarrow g_1, \cdots, g_n$	hedge unchanged
$expand$	$g = g_e \rightarrow$ $g = e \wedge \beta(e) = g_e$	$H^e \rightarrow (v{:}e, H^e)$
$collapse$	$g = e \wedge \beta(e) = g_e \rightarrow$ $g = g_e$	$(v{:}e, H^e) \rightarrow H^e$
$unnest$	$[g_r^1, g_o^c]^{c_1} \rightarrow [g_r, g_o^1]^{c_1+'+'}$	$H_1^r, \ H_{1_1}^o, \cdots, H_{1_{f_1}}^o, \cdots, H_h^r, \ H_{h_1}^o, \cdots, H_{h_{f_h}}^o \ \rightarrow$ $H_1^r, H_{1_1}^o, \cdots, H_1^r, H_{1_{f_1}}^o, \cdots, H_h^r, H_{h_1}^o, \cdots, H_h^r, H_{h_{f_1}}^o$
$nest$	$[g_r, g_o^c]^{c_1} \rightarrow [g_r, g_o^{c+'+'}]^{c_1}$	$H_1^r, H_1^{*o}, \cdots, H_n^r, H_n^{*o} \rightarrow$ $H_1^r, \ H_{1_1}^{*o}, \cdots, H_{1_{f_1}}^{*o}, \cdots, H_h^r, \ H_{h_1}^{*o}, \cdots, H_{h_{f_h}}^{*o}$ $H_1^r, \cdots, H_h^r$ are distinct $H_{i_1}^{*o}, \cdots, H_{i_{f_1}}^{*o}$ have same H^r value
$fact$	$[e_1^1 \vert \cdots \vert e_h^1]^c \wedge$ $\beta(e_i) = [g_0, g_{ir}]^1 \rightarrow$ $[g_0, [e_1 \vert \cdots \vert e_h]]^c \wedge$ $\beta(e_i) = [g_{ir}]^1$	$(v_i{:}e_i, H^0, H^{ir}) \wedge i \in [1, \cdots, h] \rightarrow$ $H^0, (v_i{:}e_i, H^{ir})$
$defact$	$[g_0, [e_1 \vert \cdots \vert e_h]]^c \wedge$ $\beta(e_i) = [g_{ir}]^1 \rightarrow$ $[e_1^1 \vert \cdots \vert e_h^1]^c \wedge$ $\beta(e_i) = [g_0, g_{ir}]^1$	$H^0, (v_i{:}e_i, H^{ir}) \wedge i \in [1, \cdots, h] \rightarrow$ $(v_i{:}e_i, H^0, H^{ir})$
$ojoin$	$\beta(e_a) = [e_0^{c_{01}}, e_x^{c_x}]^1 \wedge$ $\beta(e_b) = [e_0^{c_{02}}, e_y^{c_y}]^1 \rightarrow$ $\beta(e) = [e_0^{c_{01}+c_{02}+'+'},$ $e_x^{c_x+'?'}, e_y^{c_y+'?'}]$	$(H^{*01}, H^{*x}) \wedge (H^{*02}, H^{*y}) \rightarrow$ H^{*01}, H^{*x}, H^{*y} if $H^{*01} = H^{*02}$ or $H^{*01}, H^{*x}, H^{*02}, H^{*y}$ if $H^{*02} \neq H^{*02}$
$split$	$\beta(e) = [e_0^{c_0}, e_x^{c_x}, e_y^{c_y}] \rightarrow$ $\beta(e_a) = [e_0^{c_0}, e_x^{c_x}]^1 \wedge$ $\beta(e_b) = [e_0^{c_0}, e_y^{c_y}]^1$	$H^{*0}, H^{*x}, H^{*y} \rightarrow$ $(H^{*0}, H^{*x}) \wedge (H^{*0}, H^{*y})$
$setm$	$g^c \rightarrow g_{c_1}$	$H_1, \cdots, H_m \rightarrow H_1, \cdots, H_n$ if $c_1 =' +' \vert '*'$, $n = m$; else $n = 1$

4 Implementation Remarks

In this section, we comment on the implementation of the restructuring operators. The DTD restructuring parts of the operators are expected to be implemented in a graphical tool so that restructuring can be defined by dragging elements and multiplicities around in the tool. The tool then generates corresponding XQuery code to perform document restructuring. We use an example to show the XQuery code for the *mout* operator. Let $\beta(A) = [B^*, C^*]$ and $T_i = (\sqcup{:}A\ (\sqcup{:}B)\ (\sqcup{:}B)\ (\sqcup{:}C)\ (\sqcup{:}C)\ (v{:}C))$. The $mout([B^*, C^*]) \rightarrow \beta(A) = [B^*, C^*]$ and $T_o = (\sqcup{:}A\ (\sqcup{:}B)\ (\sqcup{:}C)\ (\sqcup{:}B)\ (\sqcup{:}C)$. The code for this operation is

```
<A>{ for $x at $i in  doc("t1.xml")/A/B,  $y at $j in  doc("t1.xml")/A/C
where $i = $j   return ($x, $y) }</A>
```

This example is simple and not general. However, we claim that a general implementation of the operators is possible because "XQuery is Turing complete" [4].

5 Conclusion

Restructuring source data is an important part of data integration. In this paper, we formally defined a set of restructuring operators which aim to restructure a DTD to any other DTD if filtering, update, and join are not considered. To clarify the semantics of the operators, each definition contains exact processes on how a document should be transformed to conform to a restructured DTD. Following these definitions and as future work, we will investigate information preservation properties of these operators and consider how a sequence of such operators can be automatically derived when an input and output DTDs are given so that the the input DTD can be automatically restructured to the output DTD.

References

1. Latha S. Colby. A recursive algebra for nested relations. *Information Systems*, 15:567–82, 1990.
2. Martin Erwig. Toward the automatic derivation of xml transformations. *LNCS 2814 - ER 2003 Workshops ECOMO, IWCMQ, AOIS, and XSDM Proceedings*, pages 342–354, 2003.
3. Peter Fankhauser, Mary Fernndez, Ashok Malhotra, Michael Rys, Jrme Simon, and Philip Wadler. The xml query algebra. *W3C Working Draft - http://www.w3.org/TR/2001/WD-query-algebra-20010215*, 2001.
4. Mary Fernandez and Jerome Simeon. Growing xquery. *ECOOP 2003*, pages 405–430, 2003.
5. H. V. Jagadish, Laks V. S. Lakshmanan, Divesh Srivastava, and Keith Thompson. Tax: A tree algebra for xml. *8th International Workshop on Databases and Programming Languages (DBPL)*, 2001. September, Rome.
6. Jixue Liu, Ho-hyun Park, Millist Vincent, and Chengfei Liu. A formalism of xml restrucuring operations. *http://www.cis.unisa.edu.au/~cisjl/publications/restru-with-append.pdf*, 2006.
7. Carlo Sartiani and Antonio Albano. Yet another query algebra for xml data. *IDEAS*, pages 106–115.
8. H. Su, H. Kuno, and E. A. Rudensteiner. Automating the transformation of xml documents. *WIDM*, page 6875, 2001.
9. Lucas Zamboulis. Xml data integration by graph restructuring. *BNCOD*, pages 57–71, 2004.
10. Lucas Zamboulis and Alexandra Poulovassilis. Using automed for xml data transformation and integration. *Third International Workshop on Data Integration over the Web (DIWeb)*, 2004. Latvia.

FTT Algorithm of Web Pageviews for Personalized Recommendation

Shen Yunfei[1], Qin Zheng[1], Yuan Kun[2], and Luo Xiaowei[3]

[1] Software School of Tsinghua University, Beijing, 100084, China
utcloud@163.com, qingzh@mail.tsinghua.edu.cn
[2] Research center of University of Science and Technology of China,
Anhui Hefei, 230062, China
cloudskysea@163.com
[3] Department of Construction Management, Tsinghua University,
Beijing 100084, China
xiaowei99@mails.tsinghua.edu.cn

Abstract. As the need for personalized services sharply increases caused by the booming of Internet, Web-based data-mining is becoming a valuable sources of thoughts and theory to satisfy the personalized system function. The characters of personalized data-mining is reviewed and discussed in the beginning, and then an innovative algorithm (FP-Tree time－validity algorithm) of Web pageviews, based on personalization, is raised. More authentic information can be efficiently got by adding time-validity coefficient to FTT-Tree storage structure to implement increment mining.

Keywords: Data mining, Web mining, Personalization, Association rule, Time-validity.

1 Introduction

While Internet rapidly expands, lots of Websites turns to offer exact information to specific users conveniently and speedily through re-organizing their information service pattern. Correlation rules, one of the data-mining measures, acts important roles during the process of searching users' interest hidden behind data and correlations between them to forecast the trend of the development.

2 Literature Reviews

R. Agrawal and others gave the definition[1] in 1993, making it possible to search association rules in mass transaction data. On the basis of association rules, they introduced a more efficient Apriori Algorithm in 1994[2]. But Apriori Algorithm needs to scan the database many times and may bring lots of candidate sets. To solve the problem, Han and others[3] introduce Fp-Tree structured algorithm in 2000, which improves the data-mining efficiency obviously.

Universities and academic institutes all over the world proposed many helpful ideas in research on web log. PageGather[4] selects unlinked but attractive web pages

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 133 – 139, 2006.
© Springer-Verlag Berlin Heidelberg 2006

for clustering by analyzing users' behaviors; AVANTI[5] ask users to raise their interests first, then based on the known users' information, he forecasts not only the next web page which specific user will visit, but the purpose user surf the websites.

Since traditional association rules seldom consider data time-validity while valuable information contained in data record is of great time-validity, some data's value may change as time passes. An improved data-mining algorithm （FP-Tree time—validity algorithm) based on association rules is introduced in the following paragraph. FP-Tree time—validity algorithm（FTT） uses FP-tree structure to storage frequent item sets, and with time-value coefficient added, produce personalized recommendation system with desirable time-validity.

3 Formularized Description of Web Log Data

3.1 Association

Web log data can be expressed by quadruple:

$$Z=<U,S,T,D>$$

In the quadruple, U refers to Visitor (usually represented by IP or username); S refers to previous sets of visit; T refers to time of visit; D refers to intention sets of visit. Web log session is generated by visitors and then saved with precision of seconds. Web log data is always the only one that can determine what the users' sessions contain. Via analyzing the preceding quadruples, this paper discloses the trend of users' interest changes during a specific period .

Example 1: Weblog (shown in table 1)

Table 1. Weblog Example

U	S(url)	T	D(url)
143.15.16.20	/admissions/	2002-03-11 15:8:57	/admissions/_vti_cnf/ general.asp

Definition 1. Session predication φ ---a property expression of session, which can be formulated when expressing web log's quadruple: Z=<U,S,T,D >.

Preceding process was a sample of data-mining for a single user session. Using the method [6], we can get all association rules from session files consisted of all users' sessions. All association rules applied in the personalized recommendation system have following definitions[7][8]:

Definition 2. If (d1,d2,d3,d4,d5)∈S, dj∈D, (d1,d2,d3,d4,d5) ¯ dj, dj is a forward rule with length of 1 and (d1,d2,d3,d4,d5) is a backward rule with length of 5.

3.2 Episode Rule

Web log data is composed of ordered time seqence and characterized with obvious time marker: timestamps.

Time plays an important role in the quadruple mentioned above. It takes a long time to create and improve. Web data's time-validity should be rather concerned for its dramatic changing frequency.

Definition 3. Episode, expressed by binary set α =(D, $\leq$) ,in which D is session prediction set and $\leq$ is partial Ordering meaning value varies as time passes. em with desirable time-validity.

4 Improved FTT Association Rules Algorithm

4.1 Time-Validity Model

According to the signification of time entropy, time's aftereffect decreases as time interval increases[9][10][11].

Definition 4. Time-validity :Given ordered time serials S1, S2, S3, S4, association A when S1 happens, association B when S2 happens, association C when S3 happens, time validity of association C is the strongest in the coming time S4 serial.

In figure 1, there are N point (t0,t1,t2,t3…,tn) in time axis, tn is the present point, value curve of the rule is V=e-at. It is obviously that rule's value decrease as time passes shown in figure 1.

Definition 5. Time value function of rule: The function is

$$V= e^{-a| tk-tn |} \quad (t_0 \leq t_k \leq t_n) \tag{1}$$

Definition 6. Time value coefficient of rule:

$$e^{-a| tk-tn |} \quad (t_0 \leq t_k \leq t_n) \ ,$$

maximum of which is 1.

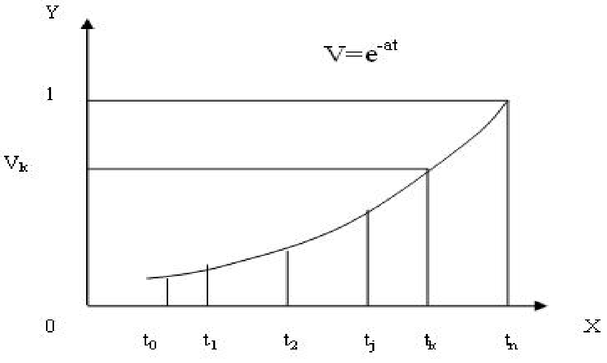


Fig. 1. Value curve of the rule

Time-validity of past rules lessens with new rules add-in, so time-validity increment is introduced.

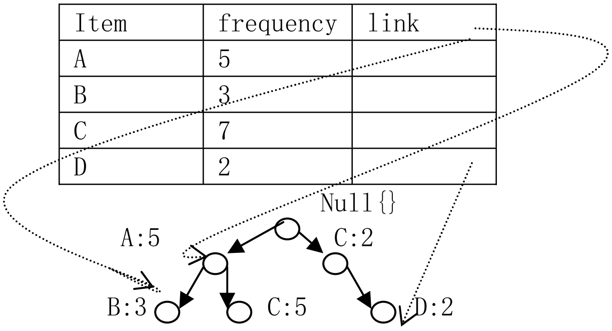
Definition 7. Time-validity increment Suppose maximum time-validity of previous rule is 1, time interval added is 1+M(0<m<1, then time value coefficient of previous rule ε becomes

$$\frac{1}{(1+m)} * e^{-a| tk-tn |} \quad (t_0 \le t_k \le t_n) \qquad (2)$$

4.2 Create FTT Algorithm Based on Web Log Data

4.2.1 Data Storage Structure

In order to lower space and time cost, FTT algorithm adopts FP-Tree storage structure to search for frequent item sets. In the structure, a root node marked with “null” is the root of tree; each node on the prefix subtree in the project involves project name and association numbers in connected paths. The nodes leads to the forward node (null if absent) and backward node (null for root node) with same value.



Item	frequency	link
A	5	
B	3	
C	7	
D	2	

Fig. 2. FTT Data storage structure

4.2.2 Algorithm Idea

Utilizing algorithm idea “k association rules on one-support count” proposed in PARM[12] algorithm , FTT algorithm gets frequent items set first based on the association rule of personalized recommendation system.

Overall days of data storage in database can be calculated by scanning database, then we can figure out time parameter with formula $V=e^{-a| tk-tn|}$.

Time.validity(ε) //calculate time value coefficient

```
Begin
N=0;
while(T!=null){ // scanning time parameter in database
n=n++;     }// calculate days
For i=0 to n;
```

(6) Calculate value coefficient in each time interval using value calculation function $V{=}e^{-a|tk-tn|}$

(7) When new date in

(8) Previous value coefficient becomes $\frac{1}{(1+m)} * e^{-a|tk-tn|}$

(9) End;

New frequent items set comes out by multiplying original frequent items set of rule and corresponding time value coefficient. Different sets can be selected out in needs of specific confidence and support.

5 Algorithm Assessment

All experiments introduced here are run on PC with 933 MHz CPU and 384M RAM, program are run in Microsoft Windows 2000 Server system.

First we take transaction data set of DEPAUL University (US) network for sample and treat them with FTT, PAPM and FP-GROWTH algorithms for test. Comparison figure of time cost using different confidence are shown in figure 3:

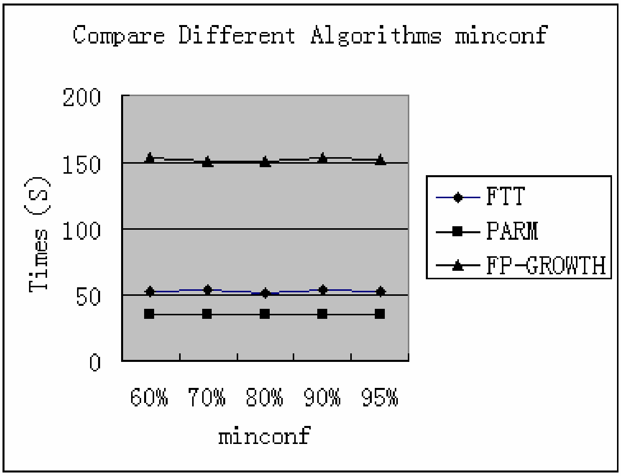


Fig. 3. Comparison different confidence

From the figure, we can find that time cost is insensitive to change minimum confidence. In the aspect of running speed, PARM is supreme, FTT medium and FP-Growth lowest.

Next we use FTT and PARM algorithm to deal with the data set of general operation information website, then we use the calculating result for personalized recommendation. The website owns 234 URL, 192 users, logged in with true name. We take 176 users with over one year using history for sample to do the analysis. Analysis results are used for personalized pageviews recommendation and test for two weeks. We do survey on the 176 users and get 145 valid questionnaires. Result is shown in Table 2:

Table 2. Result for personalized recommendation

	No recommendation	PARM	FTT
good	10%	30%	56%
commonly	30%	52%	34%
bad	60%	18%	10%

In figure 3, users' satisfaction leveled up after personalized pageviews recommendation, and effect of FTT algorithm is more distinct than PARM. For web data with notable time-validity, FTT can reflect time-validity of association rules well, resulted in better performance and authenticity of personalized pageviews data-mining.

Increment mining method is taken for time-value coefficient mining. Data calculation burden for each increment is only 1/(1+m) , lowering the server's pressure. FTT algorithm can be used for real-time and parallel computation, so it is very suitable for real-time data like weblog.

6 Conclusion and Discussion

An association rule algorithm is proposed aim to improve the data-mining of Web pageviews personalized system in this paper. The feature of it is "time value coefficient of association rule", using FP-Tree storage structure to figure out time value of association rules. The algorithm can calculate the confidence of specific association rule at present more efficiently, based on which we can forecast time value in the coming period exactly.

Personalized Web Service is an extremely wide scope to do research on. Much further work can be done focusing on the following topics: improving website algorithm, research on user behaviors model and service quality assessment model. All the topics mentioned above should be researched on.

References

1. Agrawal. R,Imielinski T,Swani A.Mining Association Rules Between Sets of Items in Lare Databases.In:Proceedings of Acm SIGMOD Coference on Managerment of Data, (1993) 207-216.
2. Agrawal. R, SrikantFast. R.. algorithms for mining association rules in large databases.In Research Report RJ9839,IBM Almaden Research Center,San Jose,CA (1994)
3. Jiawei Han, Micheline Kamber. Data Mining: Concepts and Techniques BeiJing: China Machine Press (2001)
4. Mike Perkowitz,Oven Etzioni,Towards Adaptive Web Sites:Conceptual Framework and case study,Artificical Intelligence 118 (2000)245-275.
5. J Fink,A.Kobsa,A.Nill,User-oriented Adaptivity and Adaptability in the AVANTI. Microsoft Usability Group,Redmond,Washington,USA (1999)
6. Movasher B,Cooley R,Srivastava J,Automatic personalization based on web usage mining. Communications of the ACM, (2000) 142-151

7. Movasher B,Dai H,Luo T,et al. Efficient personalization based on association rule disvoery form web usage data. Movasher B ed. 3rd Int Workshop on Web Information and Data Managerment (WIDM 2001)[C].New York: ACM Press, (2001）9-15
8. Lin w,Alare S A,Ruiz C,Efficient adaptive support association rule mining for recommender systems, Data Mining and Knowledage Discovery, (2002) 83-105.
9. Harms, Sherri K., Deogun, Jitender S. Discovering Sequential Association Rules with Constraints and Time Lags in Multiple Sequences, Journal of Intelligent Information Systems, v 22, n 1, (2004）7-22.
10. Harms, Sherri K., Deogun, Jitender S, Tsegaye Tadesse Sequential Association Rule Mining with Time Lags Lecture Notes In Computer Science; Vol. 2366Proceedings of the 13th International Symposium on Foundations of Intelligent Systems(2002）432 - 441
11. Wang Xiaoguo, Huang Shao kun, Zhu Wei, Time-validity in miningassociation rules, Computer Applications, （2005）
12. Yan Ying, Wang Daling, Yu Ge, Association Rules Minging Algorithm of Web Pageviews for Personalized Recommendation, Computer Engineering,（2005）

D-FOAF: Distributed Identity Management with Access Rights Delegation

Sebastian Ryszard Kruk[1], Sławomir Grzonkowski[1,2], Adam Gzella[1,2],
Tomasz Woroniecki[1,2], and Hee-Chul Choi[3]

[1] Digital Enterprise Research Institute,
National University of Ireland, Galway, Ireland*
firstname.lastname@deri.org
[2] Faculty of Electronics, Telecommunications and Informatics,
Gdansk University of Technology, Poland
[3] DERI Seoul National University, 28-22 Yeonkun-Dong,
Chongno-Ku, Seoul 110-749, Korea

Abstract. Todays WWW consists of more than just information. The WWW provides a large number of services, which often require identification of it's users. This has lead to the fact that today users have to maintain a large number of different credentials for different websites - distributed or shared identification system are not widely deployed. Furthermore current authorisation systems requires strict centralisation of the authorisation procedure - users themselves are usually not enabled to authorise their trusted friends to access services, although often this would be beneficial for services and businesses on the Web.

In this article we present D-FOAF, a distributed identity management system which deploys social networks. We show how information inherent in social networks can be utilised to provide community driven access rights delegation and we analyse algorithms for managing distributed identity, authorisation and access rights checking. Finally we show how the social networking information can be protected in a distributed environment.

1 Introduction

The Internet provides a large number of different services. Usually services require authentication of their customers. Two most common examples that require user authentication are access control to services or resources, and personalisation of aservices. The usuability of services suffers greatly from the fact the usually no single sign in facility is available.

* The support of Enterprise Ireland, through its Informatics Commercialisation initiative, for the eLITE Industry Led Research Project in eLearning is gratefully acknowledged. This material is also partially based upon works supported by KBN, Poland under grant No. 4T11C00525, and partially by a grant of the Interoperable EHR Research and Development Center(A050909), Ministry of Health & Welfare, Republic of Korea. The authors would like to acknowledge Stefan Decker, Marco Neumann, John Breslin, the DERI Semantic Web Cluster and the Corrib.org working group for fruitfull discussions.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 140–154, 2006.
© Springer-Verlag Berlin Heidelberg 2006

The proliferation of Internet services introduces many problems like no single identity for Internet users or no scalability in trust and access rights management. Some of those problems has been so far addressed in a number of ongoing projects.

The main difference between the Internet and real world services are authorisation procedures. In the real world each person has a single identity expressed with a number of credentials like ID card, passport or driving license. This allows real world service providers to easily confirm the authenticity of the presented credentials In the Internet, each user has to deal with a number of identities with different credentials like login-password pair. Since the is no notion of single identity service providers are usually inclined to introduce new credentials for each user. As a result the trust to each user is build within each service separately.

Approaches like Microsoft Pasport [9], Sxip [21] or Liberty Alliance Project [14] are aiming to provides a solution to the single-sign-on problem. Due to various problems none of those projects has been widely adopted by service providers so far, making them useless for the majority of Internet users with the ever growing number of service.

Most of online services are usually based on very simple user profile management implementations that do not address problems stated above. Access rights are based on predefined, fixed list of groups and neither allow finer granularity nor trust delegation.

The notion of social networking emerged in the Internet with online community portals like Orkut that allow users to control access to the information based on the structure of the social network. Each user can restrict access to some parts of his/her profile information delegating trust within given number of degrees of separation.

1.1 Contribution and Paper Overview

In this paper we present an identity management solution based on social networks. Each user has a control on his/her profile and social networking information. Access rights are based on the structure of the social network and thus very fine grained by introduction of notion access rights delegation (see section 3). Further on, we extend this solution to a distributed identity management system where only a single registration is required within the network of the user profile management systems (see section 4). We present how the sensitive information from the perspective of the user and access rights management can be protected. Finally, we present the FOAFRealm system that implements presented solutions and utilises FOAF metadata to allow exchange of the profile information with other systems (see section 5).

2 Use Case Scenario

One of possible scenarios where both distributed identity and the trust delegation is utilised is W3C information management. W3C consists of a growing number of member organisations. Each W3C Member has one Advisory Committee Representative (AC Rep). This person knows enough about the Member organisation's structure and forwards detailed technical reviews to the proper person. The AC Rep receives official notices from W3C. Acting as a gatekeeper, the AC Rep responds to, or delegates response to W3C Calls. The AC Rep appoints participants in W3C Working Groups.

Trust Delegation. When AC Rep has to grant access to some W3C services or resources, he/she needs to either add given person to an access control list or add this person to a group that already has access to the resource. In the constantly growing, evolving and changing research organisation managing access rights in that way maybe time consuming.

In this section we describe how AC Rep could delegate access rights without constant alteration of the ACLs or access groups.

AC Rep can define access rights group as a subgraph of social network within 2 degrees of separation from him/her. This allows his/her direct collaborators to delegate the access rights to W3C resources and services one step forward in the social network (see Fig. 1). This way AC Rep does not have to alter the access rights list for every new member. It is enough when at least one of existing members establish friendship relation. The new member cannot delegate the access rights any further, though.

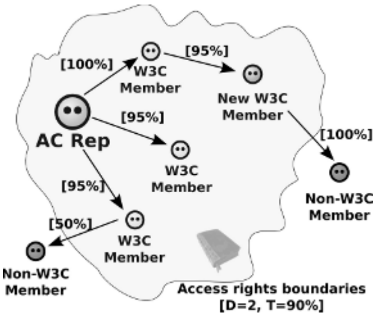


Fig. 1. W3C Scenario - Access rights delegation within the community

Many W3C Member organisations can take part in different W3C Working Groups. Access rights delegation based on the friendship relations may introduce security threats, by allowing people from different working groups to access resources allowed to other working groups. People affiliated with W3C Member can defined their friendship relations within *working group contexts.* But do not share access rights beyond working groups even though some of them stay in the direct friendship relation (see Fig. 1).

Distributed Identity Management. The identity management based on social networks provides a solution for fine granularity in access rights and trust delegation management. Some of the communities are spread across the number of different web applications, and many members very often belong to more then one. In our scenario, W3C consists of many independent member organisations that have their own identity management systems.

In today's world an typical user needs to remember a number of login-password credentials to access all of his/her accounts. When it comes to operate within the context of W3C people affiliated with member organisations have to manage additional account(s) to access the W3C resources and services. To ease the burden of handling multiple credentials and many friendship lists within different communities a distributed community driven profile management (see Def. 4) can be established across a number of different web applications.

Once all W3C Members agree on distributed identity management introducing new member organisations or new people affiliated with existing W3C Members will not force W3C to create and maintain new accounts. Additionally, there will be no need to add new access rights as they will be delegated into the local access control systems based on social networks.

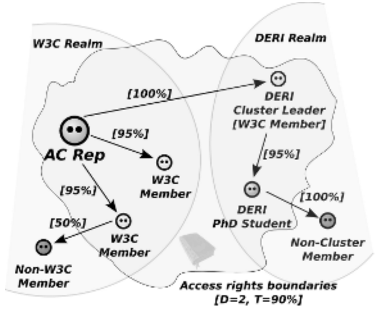


Fig. 2. W3C Scenario - Distributed identity management

3 Community Driven Access Rights Delegation

3.1 Social Networks as a Mean to Delegate Trust

In the contemporary Web 2.0 - full of wikis and community portals like Orkut [10] or LinkedIn [8] - wide community activity is perceived as a must for successful development of almost any Internet undertaking. By exploiting existing social networks to define access rights a system can easily evolve and eventually reflect the state of the real world.

Social networks driven identity management system (see Def. 1) defines access rights in terms of friendship relations between users. Friendship relation can be naively modeled with a digraph, where a direct link from A to B means 'A knows B' [39].

Definition 1 (Community Driven Access Control). *The service S that implements identity management based on social networks UPM_{SN} provides the community driven access control over resources $\{r : r \in R_S\}$ $\iff$ the changes introduced to the social network reflect the effective access rights $ACL(r)$ to the resource r.*

3.2 Going Beyond Friendship Digraph

The simple digraph representation does not cope with an important features presented in the real social networks – quality and context of friendship relation. To model social network more thoroughly each relationship can be annotated with metrics (see Def. 2) defining how long the friendship lasts, frequency of meetings, average time spent together. For example Orkut [10] lets choose from few predefined levels of friendship (like *haven't met* or *good friend*). Though these examples give absolute and comparable numbers, they usually can not be used to measure user's real feeling about particular relationship. Smoothing each context scale (e.g. from 0% to 100%) helps to describe original connection more precisely.

Definition 2 (Friendship Level Metric). *Each friendship relation $r \in R_{SN}$ between social network member $m_A \in M_{SN}$ and member $m_B \in M_{SN}$ can be annotated with a quality measure $FLM_{context}(m_A, m_B) \in < 0, 1 >$ representing friendship level metric within given context.*

3.3 Calculating User Rights on the Social Network

Definition 3 (Social Networked Access Control List). *Access control list $ACL_{SN}(m, d, l : m \in M_{SN}, d_{max} \in D_{SN}, flm_{min} \in FLM_{SN})$ defined within user profile management system based on social networks defines access rights delegation within a maximal distance $d_{max} \in D_{SN}$ and a minimal friendship level metric $flm\_context_{min} \in FLM_{SN}$. Both values are computed across the social network SN from the one member $m \in M_{SN}$ to the member requesting access to the resource.*

One of primary functions of many web applications is to assure access rights control to particular resources defined with access control lists (see Def. 3). Community driven access control system takes into account not only direct friends of the resource's owner but a whole social network. One of possible scenarios is when someone would feel that a very good friend of his/her very good friend is more trustworthy than a direct colleague he/she barely knows (see Fig. 1).

A person is granted access to a resource when the friendship level and the distance between the resource owner and the service requester meet required constraints. Distance is the length of the shortest path from the owner to the requester. Final friendship level is computed by multiplying all metric values (which are all $\in < 0, 1 >$) on a path from resource owner to the requester (highest found product is taken). Access right can be delegated further only if other requesters conform to the given distance and friendship level constraints.

To find exact values of distance and friendship level a slightly modified Dijkstra algorithm [18] can be used. Although the Dijkstra algorithm has been proved to be quite efficient, operating on an enormous social network can introduce some scalability problems. However, finding the exact values of distance and computed friendship level is not required in the context of checking access rights. To check access rights to a resource the algorithm has to find whether the distance value is lower and the friendship level is higher than the given constrains. The modification made to the Dijkstra algorithm makes it stop calculations as soon as it finds 'yes or no' answer whether to grant access - without calculating precise values. Such an optimisation often saves a lot of time during the authorisation procedure.

4 Distributed Identity Management

The identity management based on social networks introduced in previous section (see Def. 1) provides a solution for fine granularity in access rights and trust delegation management. Some of the communities are spread across the number of different web applications, and many members very often belong to more then one. To ease the burden of handling multiple credentials and many friendship lists within different communities a distributed community driven profile management (see Def. 4) can be established across a number of different web applications.

Definition 4 (Distributed Community Driven Identity Management). *A federation of interlinked user profile management services based on social networks* $\{upm : upm \in UPM_{SN}\}$ *creates a distributed community driven identity management consisting of independent profile management services cooperating in authorisation and access rights calculating procedures.*

4.1 A Remedy for Multiple Accounts in the Federation of Services

User that has an account in one of the member services, called registration server [21,38] can easily log into the other member services of the distributed community driven identity management. The user has to remember one identity credentials representing his/her indentity while the system will perform distributed authorisation [42,17] algorithm (see Fig. 3).

```
Require: $userName \neq null$ and $password \neq null$
Ensure: $auth_{result} \in \{true, false\}$
  $auth_D \Leftarrow$ perform local authentication
  if $auth_D \neq true$ then
    if $userServer \neq null$ then
      $auth_D \Leftarrow$ authenticate directly on user's server
    end if
    if $auth_D \neq true$ then
      $resultTable \Leftarrow$ perform query in network
      for $elem \in resultTable$ do
        if $elem[result] = true$ then
          $auth_D = true$
        end if
      end for
    end if
  end if
  return $auth_D$
```

Fig. 3. User authorisation in distributed environment

4.2 Protecting Social Network from Unauthorised Alterations

A social network and a distributed profile management system must be protected from many threats. The threats can be divided into several categories [20] like human-related, cookies-related or fundamental problems. Unauthorised alterations of the profile information are one of the fundamental ones.

Access rights definition in community driven access management (see Def. 1) is based on the structure of the social network. Therefore, social network information has to be especially protected by identity management system.

The improved security of the distributed social network [23,29] is introduced by signing local social networks (see Def. 5) with the private key [40] issued by the registration server for the each user.

Definition 5 (Signature on the Local Social Network). *Each integral part of the social network $sn(m, s) \subset SN$ from the perspective of the member $m \in M_{SN}$ hosted by the service $s \in S_{SN}$ is accompanied with signature created with private key at the registration server $RS(m)$.*

The signature is checked every time the social network information is accessed. The registration server RS_{SN} is responsible for generating signatures for other federated services, protect the private key information and host the public keys.

4.3 Calculating User Rights on the Distributed Social Network

To allow user to access protected resources, the service has to check the presented credentials and confirm that the user conforms to the given access control list restrictions. In other words, the service has to check if distance and friendship level meet required constrains.

Distance and friendship level metrics computations are executed each time user wants or simply attempts to access the protected resource. The process of calculating user rights in distributed network is complex, and consist of three general steps:

Step 1. System utilises the modified Dijkstra algorithm [18] to compute distance (or friendship level) between the users. In the first step of the distributed computing, the algorithm is executed at the local service (see Fig. 4). If local information conforms to the boundaries like maximal distance or minimal friendship level the algorithm terminates with success, otherwise it continues to the next step.

Step 2. In the second step, request is dispatched to each node and local computation is performed separately on each host in distributed social network. If any of the services can provide positive answer than the result is sent back to the service initiating the process and algorithm terminates.

Step 3. It might be assumed that close friendships are defined within one community managed by one of local authorisation services. It is also very possible that two people are connected through some other ones with their profiles on other nodes in the network. In this case, in order to compute distance between two people, system builds new digraph using information gathered from all hosts in network. When new digraph is created, Dijkstra algorithm is used to compute distance and friendship level metrics.

```
**Require:** $userA \neq null$ and $userB \neq null$ and $maxDist \in< 0, \infty >$ and
  $minLevel \in< 0, 1 >$
**Ensure:** $dist_{result} \in< -1, \infty >$ and $level_{result} \in \{-1\} \bigcap < 0, 1 >$
  $dist_D, level_D \Leftarrow$ perform modified Dijkstra's algorithm
  **if** $dist_D < 0$ or $dist_D > maxDist$ or $level_D < minLevel$ **then**
    $dist_D, level_D \Leftarrow$ retrieve metrics from local cache
  **end if**
  **return** $dist_D, level_D$
```

Fig. 4. Compute distance locally

```
**Require:** $dist_{res}, level_{res} \leftarrow performLocalDijkstra(gatheredDigraph)$
**Require:** $dist_{res} \in< 0, +\infty >$ and $level_{res} \in< 0, 1 >$
  $path_{new} \leftarrow$ sequence of foaf:knows triples
  **for all** $nodes \in path_{new}$ **do**
    notify user's registration server that user is cached
  **end for**
  $localCache \leftarrow path_{new}$
```

Fig. 5. Creating local cache of social network

4.4 Creating and Maintaining Local Cache of Social Network

The third step of user rights' computing can result in a huge digraph and expensive overload of the network. To perform the third step as rarely as possible a caching algorithm must be introduced (see Def. 6). The goal is to remember the result of the complex distance computing. Remembering all information gathered from other services would provide a lot of redundancy and could result in data inconsistency. The local cache keeps only paths between two nodes in the digraph D_{SN} which could be used in the first or the second step of distributed user rights computing.

Definition 6 (Local Cache of Social Network). *Each UPM_{SN} maintains a local cache of social network $LCSN$ consisting of some edges $r \in R_{SN}$ between vertexes $m(r)_A \in M_{SN}$ and $m(r)_B \in M_{SN}$ in the digraph model of social network D_{SN}.*

System creates a cache (see Fig. 5) by adding new paths to local store. Registration servers of all users that were represented by outgoing vertexes in the added path, are notified about the caching procedure. If some friendship information about the user has been changed, RS_{SN} sends update notification to services that maintain the cached information. The service that receives this notification invalidates cached path starting from the node representing the user on whom the information has been changed.

5 D-FOAF - A Distributed Identity Management System on Social Networks

The concept of a distributed identity management system has been implemented in the FOAFRealm project [4,31]. FOAFRealm delivers a plug-in for Tomcat [12] JSP container and utilises FOAF [19] metadata extended with concepts required by distributed user profile management on social networks. The main feature of FOAFRealm is the implementation of `org.apache.catalina.Realm` and `org.apache.catalina.Valve` interfaces that introduce the concept of *Community Driven Access Control* (see Def. 1) and *Distributed Community Driven Identity Management* (see Def. 4) to J2EE web applications. The use of FOAFRealm core features like authorisation and access rights management is transparent to the web application builder. FOAFRealm encodes access control definitions in a form of literals that are understood by Tomcat as realm group definitions but are processable by FOAFRealm. Example 1 shows how the *Social Networked Access Control List* (see Def. 3) is encoded in FOAFRealm.

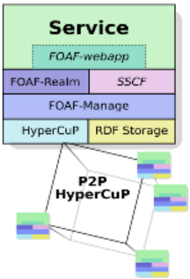


Fig. 6. Architecture of the D-FOAF system

Example 1. ACL restricting access to a resource to the network of people that are within 3 degrees of separation from the user `sebastian.kruk@deri.org` and whose trust level computed across the social network is above 50%, can be encoded in FOAFRealm as `F[mailto:sebastian.kruk@deri.org]3,5`, where 3 stands for 3 degrees of separation and ,5 represents the 50% minimal trust level.

5.1 Architecture

D-FOAF, Distributed FOAFRealm, utilises the HyperCuP P2P infrastructure to connect and exchange information between FOAFRealm instances. There are four major features supported by D-FOAF:

- Distributed user authentication (see section 5.2)
- Distributed identity management (see section 4)
- Secure distributed computing of distance and friendship level between users (see section 5.3).
- Social semantic collaborative filtering [32]

The current implementation of FOAFRealm consists of four layers (see Fig. 6):

- The distributed communication layer provides access to a highly scalable HyperCuP [37] Lightweight Implementation [5] of a P2P infrastructure to communicate and share the information with other FOAFRealm implementations.
- FOAF and collaborative filtering ontology management. It wraps the actual RDF storage, providing simple access to the semantic information from the

upper layers. The Dijkstra algorithm for calculating distance and friendship quantisation is implemented in that layer.

– Implementation of the `Realm` and `Valve` interfaces to easily plug-in the FOAFRealm into the Tomcat-based web applications. It provides authentication features including autologin based on cookies.
– A set of Java classes, tagfiles and JSP files plus a list of guidelines that can be used while developing a user interface in personal web applications. This layer includes general user interface implementations for user profile management, social semantic collaborative filtering and multifaceted browsing.

5.2 User Authentication in D-FOAF

To provide a single registration feature in the whole federation of FOAFRealm services (see Def. 4), D-FOAF performs a distributed authentication algorithm (see Fig. 3). When a user logs in for the first time, the service locates his/her registration server by sending a registration server discovery broadcast query over the HyperCup P2P network. Once the location of the registration server is found a local user profile is extended with the triple <`user_mbox`> <`foaf:seeAlso`> <`registration_service_uri`> indicating the location of the registration server to speed up authentication operations in the future. Authentication responsibility is later delegated to the user's registration server, which answers with the user's profile upon successful registration, or indicates that the supplied credentials are wrong.

5.3 Distance and Friendship Level Metrics Computing in D-FOAF

Computing distance and friendship level over a distributed RDF is required for evaluating user access rights, and is probably one of the most complex algorithms in D-FOAF. The system has to cope with a variety of problems. The problem gets less trivial when the FOAF graph is distributed among many services consisting the D-FOAF network. The distances computation is performed in three steps implementing the algorithm defined in section 4.3:

Step 1. A single instance of FOAFRealm implements the modified Dijkstra algorithm to compute the distance and the friendship level between users. Computations are performed on the local FOAF database.

Step 2. The distance and friendship level computation algorithm is performed on each node of the D-FOAF network independently. The query is send as a broadcast on the HyperCuP P2P backbone of the D-FOAF network.

Step 3. The system has to gather all the information, required to compute the distance into one place - the FOAFRealm instance that invoked the query. The complete information about the profile of the first user is retrieved. Next all <`foaf:knows`> triples describing direct friends of this person are gathered with the HyperCuP broadcast. Local server builds temporary FOAF database and performs standard local computation together with retrieving missing <foaf:knows> profile information on demand.

Caching. The third step might generate a huge RDF graph and expensive overload of the network with broadcast messages. The caching feature has been implemented in the D-FOAF to address these issues. Since the original social network in each FOAFRealm node is signed by the registration servers, <`foaf:knows`> triples that builds up the cached path are stored in a separate RDF store not to weaken the previously introduced security mechanism.

5.4 Evaluation

We evaluate FOAFRealm against current distributed identity management systems. Firstly, Microsoft Passport[9] gives a simple Single Sign-On feature. Because of the centralised topology, proprietary status and very frequent bug reports, the system has not been yet widely accepted. Moreover, users do not want to share their private information to a commercial company. The solution cannot guarantee that the privacy information will not be used for illegal purpose.

To solve this privacy problem, the Liberty Alliance Project[14] suggests open specification and multiple identity providers. The more than 150 organizations are bringing together their specification. They have also added ACL features based on social relationships. The project is targeted at larger scale and more business oriented web services and thus it is used very rarely in small enterprises. Users find it hard to make their own server and they still need to relay on large organizations. Moreover, the specification needs complicated procedure to make social relationships without opening the personal information of other people.

The SXIP[21] makes a more simple solution to support privacy. It is a lightweight and open source solution. It also gives a development kit so users can make their own private servers to save their private information. Despite the fact that it is a step forward with respect to Passport, Sxip is still centralised from the perspective of the home server. However, SXIP does not give any access control list or social relation features.

We have described the identity management and social network features of FOAFRealm. It is also an open source solution and users can have their own FOAFRealm servers. However, at the current stage, FOAFRealm exports their relationship information to other FOAFRealm servers, which can be private information and future work is needed to research how to prevent abuse of sensitive data.

To futher test research presented in the previous sections we have deployed FOAFRealm in JeromeDL [7,33]. DERI has decided to use JeromeDL (and FOAFRealm) as their main digital library engine. Websites `http://library.deri.ie` and `http://library.deri.at` successfully serve digital publications, offering distributed FOAFRealm profiles management, fine grained access control lists and semantic searching. FOAFRealm features showed to be useful for majority of users and most of them quickly adopted sharing bookmarks with friends.

6 Related Work

Our work not only introduces an interesting approach to common problems but also integrates several existing concepts. Two most fundamental are user management and social networks research areas.

6.1 User Management

Project Integration Architecture [28] researched by NASA, provides a distributed user management and access control lists. Problems of the security were considered and described for the whole process of authentication [27]. The solution, which was implemented in CORBA [2], is based on distributed lock management [26] and deadlock detection. Unfortunately, the system does not support any semantic user profile description like FOAF.

The EMBASSI [22] project propose an original approach to distributed user profile management which uses agent based architecture. The system divides user profile into two types - the personal generic user data and domain values that are relevant for specific environment. It has been shown that this approach leads to a compound set of generic user variables and it can meet the requirements for different application areas.

Identity 2.0 [6] is a protocol for exchange of digital identity information. The general idea is to provide users with more control over what others know about them. The next version of the system mentioned above - Sxip [21] will provide increased anonymity for users. Furthermore, it will be possible to adjust security needs to specific site.

MyProxy Credential Management Service[36] initiative has already solved the problem of managing different user accounts. But the work was conducted in the context of Grid and the users are not enabled to take advantage of existing social networks and semantic user profile description.

The SD3[25] is a distributed trust management system that introduces high-level policy language. The system utilising groups and permissions instead of access control lists and social networks and that is the main difference between this project and D-FOAF.

An interesting approach was proposed in PeerTrust Project[1], which concerns a decentralised Peer-to-Peer electronic community. The important contribution of these authors is to build a trust model based on only three factors: the amount of satisfaction established during peer interaction, the number of iterations between peers and a balance factor for trust. And the trust model is the main difference in comparison with FOAFRealm system.

The idea of distributed user profile management become more and more popular it results in projects developed by open source community. Drupal [3] offers distributed authentication module and Single Sign-On feature. XUP [13] takes advantage of XML format which holds user account information. This issue competes with the W3C FOAF [19] metadata recommendation.

6.2 Social Networks

The six degrees of separation [35,30] theory began the research and development of social networks. The number six derives from an experiment performed in 1967 by social psychologist Stanley Milgram [34].

Because the Milgram's experiment had been rather small, it was questioned. As a result some sociologists [41] recruited over 60,000 participants from 166 different countries and they performed tests on the Internet environment.

The first website called *HotLinks* which utilised the concept of the six degrees of separation was published in 1998, and was available for four years. Then, the members were moved to Friendster [16] network, which was founded in 2002. Since winter 2002 Friendster network is becoming more and more popular. There are more than 21 million members at the moment.

Nowadays, there are a few dozen networks that take advantage of six degrees phenomena. They differ in many ways. For example, Hungarian WIW [15] and Orkut [10] projects require an invitation in order to join the network, which guarantees that at least one relationship with community for new members, while it is not necessary in Friendster mentioned above. In addition, we noticed recently a large grow of business oriented networks, like e.g. LinkedIn [8] and Ryze [11], that manage professional contacts, enabling users to find employer or employee.

Complexity[39] is an on-Line journal. An special issue published in August 2002 was dedicated to the role of networks and network dynamic. Although, the focus was on showing complexity for different levels of network architecture, a large part of the journal was related to social networks. The mentioned issues were helpful in comprehension of network-based analyses and explanations.

The scope of social networks is much wider. Recently, the idea was adopted in order to protect from spam, which becomes such a ubiquitous problem. Introducing reputation networks and taking advantage of Semantic Web, TrustMail project [24] extends the standard social network approach. Moreover, various algorithms were considered and a prototype email client was created and tested. It resulted in highly accurate metrics. Additionally, valid e-mails from unknown users can be received, because of connection in the social network.

7 Conclusions and Future Work

We introduced the identity management based on social networks. We showed how utilising of the social networks in the identity management systems can reflect in high granularity and scalability of the access control features providing notion of access rights delegation. We detailed algorithms for the distributed identity management that have been implemented in the FOAFRealm/D-FOAF project presented in this article.

Although the FOAFRealm system presents a complete solution for distributed identity management based on social networks, there is number of issues that are being implemented at the moment. The access rights delegation based on the social network information and trust levels has been so far tackled within a single context. Further research on multiple contexts of trust levels and distributed trust computation will be carried on. The idea of single identity registration can only be realised when a lot of online services can use or connect to the D-FOAF network. To make that possible, implementations for other platforms like .NET or PHP will be provided in the future. In addition the third step of evolution of FOAFRealm system, called DigiMe, has been initiated. The goal of DigiMe project is to deliver a complete solution for mobile devices. This solution will not only provide access to existing D-FOAF networks but provide users with better control over their profile information. DigiMe will enable users to store this information on the mobile device. Finally, further research on algorithms for distributed FOAF computations including security, caching and replications will be continued.

References

1. PeerTrust Homepage : http://www-static.cc.gatech.edu/projects/disl/PeerTrust/.
2. CORBA: http://www.corba.org/.
3. Drupal: http://drupal.org/.
4. FOAFRealm project: http://www.foafrealm.org/.
5. HyperCuP Lightweight Implementation project: http://www.hypercup.org/.
6. Identity 2.0: http://www.identity20.com/.
7. JeromeDL project: http://www.jeromedl.org/.
8. LinkedIn: http://www.linkedin.com/.
9. Microsoft Passport: http://www.passport.net/.
10. Orkut: http://www.orkut.com/.
11. Ryze: http://ryze.com/.
12. Tomcat: http://jakarta.apache.org/tomcat/.
13. XML User Profiles: http://xprofile.berlios.de/.
14. L. Alliance and WS-Federation. A Comparative Overview. White Paper. Technical report, 2003.
15. G. bor Cs nyi and B. zs Szendroi. Structure of a large social network. 2004.
16. D. M. Boyd. Friendster and Publicly Articulated Social Networking. In *Conference on Human Factors and Computing Systems (CHI 2004)*, http://www.danah.org/papers/CHI2004Friendster.pdf, 2004.
17. D. Cvrcek. Authorization Model for Strongly Distributed Information Systems.
18. E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
19. L. Dodds. An Introduction to FOAF. http://www.xml.com/pub/a/2004/02/04/foaf.html, February 2004.
20. S. Grzonkowski, A. Gzella, H. Krawczyk, S. R. Kruk, F. J. M.-R. Moyano, and T. Woroniecki. D-FOAF - Security Aspects in Distributed User Managment System. In *TEHOSS'2005*.
21. D. Hardt. Personal Digital Identity Management. In *FOAF Workshop proceedings*, 2004.
22. M. Hellenschmidt, T. Kirste, and T. Rieger. An agent based approach to distributed user profile management within a multimodal environment. In *Proceedings of the Workshop on the Application of Semantic Web Technologies to Web Communities*, Rostock, Germany, 2003. International Workshop on Mobile Computing, IMC 2003.

23. P. Heymann. Distributed Social Network Protocol. Technical report, Duke University.
24. G. Jennifer, B. Parsia, and J. Hendler. Trust Management for the Semantic Web. In *Proceedings of Cooperative Intelligent Agents*, http://www.mindswap.org/papers/CIA03.pdf, 2003.
25. T. Jim. SD3: A Trust Management System with Certified Evaluation. In *IEEE Symposium on Security and Privacy*, May 2001.
26. W. H. Jones. Project Integration Architecture: Distributed Lock Management, Deadlock Detection, and Set Iteration. Technical report, John H. Glenn Research Center at Lewis Field Cleveland, OH 44135.
27. W. H. Jones. Project Integration Architecture: Initial Plan for Distributed User Authentication and Access Control. Technical report, John H. Glenn Research Center at Lewis Field Cleveland, OH 44135.
28. W. H. Jones. Project Integration Architecture: Application Architecture. Technical report, John H. Glenn Research Center at Lewis Field Cleveland, OH 44135, 2005.
29. R. Kaye. Next-Generation File Sharing with Social Networks. http://www.openp2p.com/pub.a/p2p/2004/03/05/file_share.html.
30. J. Kleinberg. Small-world phenomena and the dynamics of information, 2001.
31. S. R. Kruk. FOAF-Realm - control your friends' access to the resource. In *FOAF Workshop proceedings*, http://www.w3.org/2001/sw/Europe/events/foaf-galway/papers/fp/foaf_realm/, 2004.
32. S. R. Kruk and S. Decker. Semantic Social Collaborative Filtering with FOAF-Realm. In *Semantic Desktop Workshop, ISWC 2005*, 2005.
33. S. R. Kruk, S. Decker, and L. Zieborak. JeromeDL - Adding Semantic Web Technologies to Digital Libraries. In *DEXA Conference*, 2005.
34. S. Milgram. The Small World Problem. *Psychology Today*, pages 60–67, May 1967.
35. M. Newman. Models of the Small World: A Review.
36. J. Novotny, S. Tuecke, and V. Welch. An Online Credential Repository for the Grid: MyProxy. In J. Turner and R. Kraut, editors, *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10)*, pages 104–111. IEEE Press, 2001.
37. M. Schlosser, M. Sintek, S. Decker, and W. Nejdl. Ontology-Based Search and Broadcast in HyperCuP. In *International Semantic Web Conference, Sardinia*, 2002.
38. H. Shen and P. Dewan. Access Control for Collaborative Environments. In J. Turner and R. Kraut, editors, *Proc ACM Conf. Computer-Supported Cooperative Work, CSCW*, pages 51–58. ACM Press, 1992.
39. J. Skvoretz. Complexity theory and models for social networks. *Complex.*, 8(1):47–55, 2002.
40. M. Thompson, A. Essiari, and S. Mudumbai. Certificate-based Authorization Policy in a PKI Environment.
41. D. J. Watts, P. S. Dodds, and M. E. J. Newman. Identity and Search in Social Networks. *Science*, 296. no. 5571:1302 – 1305, May 2002.
42. T. Y. C. Woo and S. S. Lam. A framework for distributed authorization. In *CCS '93: Proceedings of the 1st ACM conference on Computer and communications security*, pages 112–118, New York, NY, USA, 1993. ACM Press.

Community Focused Social Network Extraction

Masahiro Hamasaki[1], Yutaka Matsuo[1], Keisuke Ishida[1], Yoshiyuki Nakamura[1], Takuichi Nishimura[1], and Hideaki Takeda[2]

[1] National Institute of Advanced Industrial Science and Technology (AIST),
Tokyo, Japan
[2] National Institute of Informatics (NII),
2-1-2 Hitotsubashi, Chiyoda-ku Tokyo, Japan

Abstract. A social networking service can become the basis for the information infrastructure of the future. For that purpose, it is important to extract social networks that reflect actual social networks which users have already had. Providing a simple means for users to register their social relations is also important. We propose a method that combines various approaches to extract social networks. Especially, three kinds of networks are extracted: user-registered *Know-link* networks; Web-mined *Web-link* networks; and face-to-face *Touch-link* networks. This paper describes the combination of social network extraction for an event-participant community. Analyses on the extracted social networks are also presented.

1 Introduction

This paper presents an integrated method for social network extraction. Social networks play important roles in our daily lives. Social networks overwhelmingly influence our lives without our knowledge of their implications. Many applications use social networks [12]. In the context of the Semantic Web, social networks are crucial to realize a web of trust, which enables the estimation of information credibility and trustworthiness [3][6]. Ontology construction is also related to social networks [9].

People pay attention to social networks not only for academic study, but also for commercial services. Social networking services (SNSs) have become popular. Friendster [3] and Orkut [4] are among the earliest and most successful SNSs. An interesting aspect of SNSs is that a users can visualize social networks (acquaintance lists) in addition to others' personal attributes (e.g., name, affiliation, hobby). Acquaintance lists reveal information about users' personalities. On the other hand, acquaintance lists are proof that acquaintances can track activity within the SNS. We can expect that the purview of acquaintances that such lists offer serves to restrict anti-social behavior within a community (e.g., assuming false names, abusive language). In fact, a famous Japanese SNS called mixi [5]

[3] http://www.friendster.com/

[4] https://www.orkut.com/

[5] http://mixi.jp/

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 155–161, 2006.
© Springer-Verlag Berlin Heidelberg 2006

has three million users; 70% of them are active users. In all, mixi has 490,000 communities (BBS), but maintaining communications on online communities presents some difficulties [4]. An SNS that manages and stores social networks can become a base of our future information infrastructure.

Currently, we are doing an *Event Space Information Support Project.* We have targeted event spaces such as expositions or conventions because such events involve rich contents and many attendees. This project is intended to activate the community by supporting real-world-based interaction using Ubiquitous and Semantic Web technologies. Ishida [5] explained that community computing includes five functions for encouraging social interaction in communities: knowing each other, sharing preferences and knowledge, generating consensus, supporting everyday life, and assisting social events. We try to realize such functions using social networks.

What is a difference between an ordinary SNS and an SNS for an event participants' community? All SNS users use SNS. Nevertheless, not every event participant uses it. Furthermore event participants' communities have already had social networks and are created in event spaces and other places. Our necessary SNS should reflect actual social networks in that community. It requires a method to obtain social networks not only from user registration but also others. We called this challenge *Community Focused Social Network Extraction.*

We propose a new method to extract social networks. It is a combination of three methods. We targeted an academic conference as the first trial community for our proposed method and developed a system that has our proposed method. We operated our system at some academic conferences. We have analyzed their respective social networks. This paper describes characteristics of three means of extraction of social networks and discusses the effectiveness of their combination.

2 Community Focused Social Network Extraction

2.1 Approach of Social Network Extraction

This section presents a summary of methods to obtain social networks. Several means exist to obtain social networks: Friend-of-a-Friend (FOAF) is a vocabulary to describe information on a person and their relation to others. We can collect FOAF files and obtain a FOAF network [2][8]. Users create both SNS data and FOAF data themselves.

On the other hand, automatic detection of relations is also possible from various sources of online information such as e-mail archives, schedule data, and Web citation information [1][14][10]. Especially in some works, social networks are extracted by measuring the co-occurrence of names on the Web using a search engine [8][7].

Another means has been explored to obtain social networks: observing persons' behaviors using ubiquitous and wearable devices [11].

Whichever method is taken for obtaining a social network, it suffers from some flaws. For example, SNS data and FOAF data, which are based on self-reporting, suffer from data bias and sparseness. Users might name some of their work acquaintances, but they might not include private friends. Some name hundreds of friends, while others name only a few. People create SNSs data by selecting registered users on the SNS and FOAF data by naming others freely. Sparsity in FOAF data is more serious than SNSs data. Automatically obtained networks, e.g., Web-mined social networks, provide a good view of prominent persons, but they do not properly record relationships of novices, students, and other "normal" people. Social networks observed using wearable devices are constrained by device-specific characteristics: they might have detection errors, limitation of detection scopes, and biased usage by users.

2.2 Our Proposed Method

This paper presents a method to extract social networks for a specific community. In this case, we target an academic conference participants' community. For extracting social network from a community, it is important to obtain cooperation from community members. For that reason, we infer that it is important to create the initial network without demanding personal information input from the user to make the whole social network useful [13]. The initial network should be modified according to user interaction to the system. For example, the initial information from the user is from the web system that has a click button to show that a user knows this person or is interested in some content, which preference might resemble the preferences of other users. Such information from web systems is added to the social network. Real-world-based user interaction information is also added, e.g., if three users used the same table together and the same demonstration was visited simultaneously by two other users.

Figure 1 shows our proposed methods. Ordinarily, a user registers a social network (1) in an SNS. We have proposed methods to extract social networks using user interaction (2) and web mining (3). Conventional methods cannot extract social networks among a community automatically and improve them using user-system interactions. We proposed a combined social network extraction method that includes many web services based on community interests and contents and onsite systems that have been deployed in real-world space for supporting mobile users in the site.

3 POLYPHONET Conference

We developed *POLYPHONET Conference* (hereafter, *POLYPHONET*) that has our proposed method. The system is a community support system whose target is an academic conference.

POLYPHONET Conference has functions as a social networking service and a conference scheduling system. A user can find research topics that a researcher is exploring or with whom she is working. In the scheduling part, a user can register interesting presentations (papers, demos and posters)

Our proposed method thereby comprises three means to extract social networks. The first is based on web mining techniques. This method can create initial networks automatically from available web information. The second is based

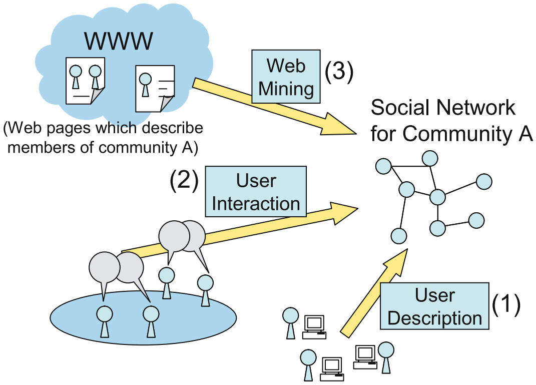

WWW
(Web pages which describe members of community A)
Web Mining
(3)
Social Network for Community A
(2)
User Interaction
User Description
(1)

Fig. 1. Proposed Method

on real-world-based user interaction in communities. Furthermore, the method captures user interactions in the conference room. The last is based on user interaction on the web system. It gathers user clicks of acquaintance buttons: it is similar to SNS. We call the first a *Web-link* and the second a *Touch-link*; the last is a *Know-link*.

A *Web-link* is extracted from the Web using Web Mining techniques. We applied the web mining method based on method [7] to extract a social network among participants. That method is based on measuring the relevance of two nodes based on the number of retrieved results obtained by a search engine query.

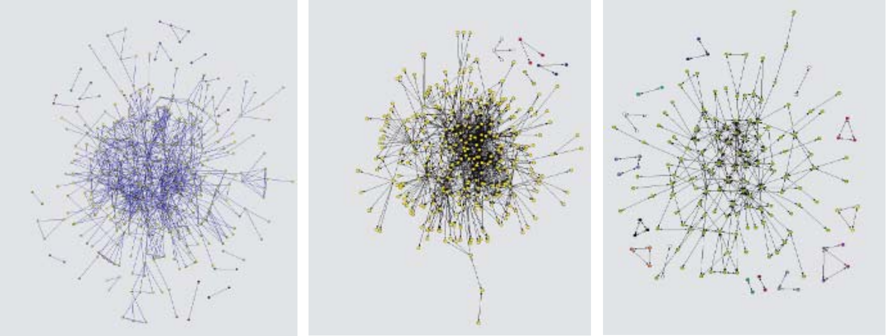
Users register a *Touch-link* via an information kiosk. We set several information kiosk in an public space and deliver each participant an ID card as a name card. They can view social networks among them and compare personal schedule if two or three participants place ID cards there together. Then the social-tie "We meet and see social networks together" is added to *POLYPHONET* automatically. It seemed to provide easy and a good balance between privacy and effectiveness because user understands how and to whom the relation will be created.

Users on the web system can register a *Know-link*. A user can make an addition to the "I-know" list when that user finds an acquaintance. At that time, the acquaintance is also added to the acquaintance's "I'm-known-by" list.

The system has a portal page that is tailored to an individual user, called *my page*. The user's presentations, bookmarks of presentations, and registered acquaintances are shown along with the social network extracted from the Web. It helps users to register *Knows-link* easily by seeing the *Web-links*.

4 Field Test

We tested our system at 17th, 18th and 19th Annual Conferences of the Japan Society of Artificial Intelligence (JSAI2003, JSAI2004, and JSAI2005) and at



Web-link Network
(node=415, edge=1049)

Know-link Network
(node=94, edge=1326)
The network is directional

Touch-link Network
(node=162, edge=288)

Fig. 2. Three Types of Social Networks

The International Conference on Ubiquitous Computing (UbiComp 2005) and analyzed obtained social networks. In this paper, we show some results of analyses on obtained networks from JSAI2005.

That conference included 297 presentations and 579 authors (including co-authors). About 500 participants joined that conference: its size was similar to that of a conference at which we tested the previous system. The system started at 5th June. In all, our system's users were 217.

POLYPHONET extracts social networks among participants using three methods. First, the system extracts them using Web mining technique as initial data (*Web-link*). Second, users can register their own social networks with our system (*Know-link*). Third, two or three users can register their face-to-face meeting through information kiosks (*Touch-link*).

As a result, *Web-link* has 484 nodes and 34,880 links. *Touch-link* has 162 nodes and 288 links. *Know-link* has 94 nodes and 1,326 links. Figures 2 show the respective networks. In these figures, the *Web-link* network threshold is controlled to reduce the number of edges and allow clear visualization.

Web-links are more numerous than others, which serves our system well. It uses *Web-links* as initial data of social networks. It is therefore desirable that every user has initial personal data. The number of users of *Touch-link* is larger than *Know-link* even though users can create *Touch-link* for only three days. It indicates *Touch-link* can provide an easy way to register social networks for community members.

About half of the *Know-links* are common with *Web-links*, indicating the initial network validity. From user log analysis, 52% of the *Know-links* are registered from a user's *Web-links* page. This suggest that at least in *POLYPHONET*, the *Web-link* contributes to set *Know-links* efficiently.

Figure 3 shows the number of Web hit (by putting a person's name to a search engine) versus the number of three kinds of links. The more authoritative people (with lots of hit count) tend to have more number of *Web links*. While

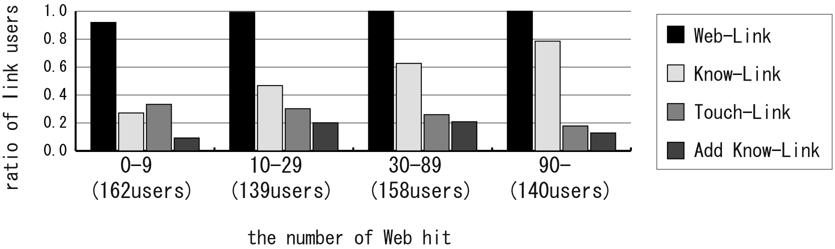


Fig. 3. Ration of Link Users at JSAI2005. 'Know-Link' means users who have undirected *Know link* and 'Add Know-Link' means users who added *Know link*.

the most authoritative people do not use *Knows links* the most; active middle-authoritative users use the most. They may know well about the community, and feel interesting. *Touch links* are used by the less authoritative users more; especially, the persons with the same level of authoritativeness are likely to have meets link. It is natural because persons who have fewer acquaintances want more acquaintances, and people are likely to meets people with the same social level.

Community focused social network extraction should reflect social network of a target community. Our target community is participants of an academic conference and it has various members. Results of this field-test shows our proposed method extracts social network from such various members.

Every *Web-link*, *Touch-link*, and *Know-link* has own characteristic even though they mean same social relationship. When we make applications with three links, we should improve a way to integrate them for each application. For example, *Web-link* is more effective when we use social network as context like a presentation page in *POLYPHONET* because *Web-link* can cover relationship of authoritative people. When we use social network for communication support, it seems that *Touch-link* and *Know-link* is more effective because they are created by active users. Especially, *Touch-link* is a key because actually a person meets others on site, the person feel easy talking again to introduce someone.

5 Conclusion

This paper presents community-focused social networks and extraction methods. We implemented our proposed method to *POLYPHONET Conference* and applied it to the academic conference. Results of field-testing show that our proposed method can realize unbiased extraction of social networks and provide a simple means to register social networks for users.

Community Focused Social Network Extraction is important for using SNS as an information infrastructure of one community. Social network is related to Semantic Web technology. If SNS is approved as an information infrastructure, the achievement of advanced information support by Semantic Web technology becomes possible, too. I hope this research becomes the help. Future studies will address network integration methods, especially how to integrate various kinds of edges, and novel applications using social networks for supporting communities.

Acknowledgments

This research has been supported by NEDO (New Energy and Industrial Technology Development Organization) as the project ID of 04A11502a.

References

1. L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.
2. T. Finin, L. Ding, and L. Zou. Social networking on the semantic web. *The Learning Organization*, 2005.
3. J. Golbeck and J. Hendler. Inferring trust relationships in web-based social networks. *ACM Transactions on Internet Technology*, 2005.
4. J. Grudin. Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM*, 37(1):99–105, 1994.
5. T. Ishida, editor. *Community Computing: Collaboration over Global Information Networks*. John Wiley and Sons, 1998.
6. P. Massa and P. Avesani. Controversial users demand local trust metrics: an experimental study on epinions.com community. In *Proceedings of AAAI-05*, 2005.
7. Y. Matsuo, J. Mori, M. Hamasaki, H. Takeda, T. Nishimura, K. Hashida, and M. Ishizuka. Polyphonet: An advanced social network extraction system. In *Proceedings of WWW2006*, to appear.
8. P. Mika. Flink: Semantic web technology for the extraction and analysis of social networks. *Journal of Web Semantics*, 3, 2005.
9. P. Mika. Ontologies are us: A unified model of social networks and semantics. In *Proceedings of ISWC2005*, 2005.
10. T. Miki, S. Nomura, and T. Ishida. Semantic web link analysis to discover social relationship in academic communities. In *Proccedings of SAINT2005*, 2005.
11. A. Pentland. Socially aware computation and communication. *IEEE Computer*, 2005.
12. S. Staab, P. Dmingos, T. Finin, P. Mika, A. Joshi, J. Golbeck, A. Nowak, L. Ding, and R. R. Vallecher. Social network applied. *IEEE Intelligent systems*, pages 80–93, 2005.
13. H. Takeda and I. Ohmukai. Building semantic web applications as information/knowledge sharing systems. In *Proceedings of End User Aspects of the Semantic Web*, 2005.
14. J. Tyler, D. Wilkinson, and B. A. Huberman. Email as spectroscopy: automated discovery of community structure within organizations. *Communities and technologies*, pages 81–96, 2003.

Behavioral Analysis Based on Relations in Weblogs

Tadanobu Furukawa[1], Tomofumi Matsuzawa[2], Yutaka Matsuo[3],
Koki Uchiyama[4], and Masayuki Takeda[2]

[1] Graduate School of Science and Technology, Tokyo University of Science,
2641 Yamazaki, Noda-shi, Chiba, Japan
furukawa@mi.ci.i.u-tokyo.ac.jp
[2] Dept. of Information Sciences, Tokyo University of Science,
2641 Yamazaki, Noda-shi, Chiba, Japan
{t-matsu, takeda}@is.noda.tus.ac.jp
[3] National Institute of Advanced Industrial Science and Technology,
1-18-13 Sotokanda, Chiyoda-ku, Tokyo, Japan
y.matsuo@aist.go.jp
[4] hottolink, Inc.,
2-11-17 Nishigotanda, Shinagawa-ku, Tokyo, Japan
uchi@hottolink.co.jp

Abstract. This paper analyze the influence of the relations among blogs on users' browsing behavior assuming that users' activities can be predict by the relations on the blog network. We define the measure for two-hop relations as a factor to influence activities, and check the correlation between them or with users' behavior by using blog data including visiting behavior. Attempting to determine the relations on which users read the blogs with interest as a helpful information for page recommendation, we conduct the experiments with a machine learning. As a result, even though the performance is not very high, we get the effective factors for prediction.

1 Introduction

Recently, Weblogs (blogs) are receiving attention as a new form of a communication tool on the WWW. Blog has characteristics include: users update their contents frequently because of ease for their management and they can debate one another about one interest topics through the functions of *comment* and *trackback*. Considering such interactive activities, we can discover the relationships among blogs (and bloggers). Relationships of users can play an important role in the Semantic Web. For example, [1] calculate the trust, one of the purpose of the Semantic Web, with a transitivity between users on social networks. Particularly, a blog is highly individualized, so in recent years it is often read with social networking services (SNSs), services that record and map human relations information, which emphasizes the effectivity of relations among blogs.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 162–168, 2006.

© Springer-Verlag Berlin Heidelberg 2006

In this paper, we notice the blog network and analyze their effects on users' browsing activities. First, we verify the hypothesis *whether users who visit blogs are strongly related*. We prepare various relations, and inspect those relations that appear to be influential. Next, we clarify *whether we can distinguish blogs that users read frequently using the relations*. If possible, the result might be useful information for building a page recommendation service. Our analyses use the database: *Doblog* [1], a blog-hosting service in Japan. We can treat users' behavior including visiting activity and analyze them. Below, we define measures of relations among blogs, check the correlations between them and users' behavior, and conduct a machine learning analysis. As a result of them, we get special effectivity of two-hop relations for activity.

The subsequent section explores related works. Section 3 explains the way of our analysis. In Section 4, we show the results. Section 5 describes analysis of the results and study with an appended experiments. We conclude the paper in Section 6.

2 Related Works

Our research treats the blog network, communities and users' activities, then we analyze the relations among those. Research on blogs are rapidly increasing in recent times, especially there are lots of ones about communities. Many studies collect a large amount of blog data and analyze it. Kumar proposes a method to identify bursty community of blogs [2]. Taking notice of the timing, this analyze the process of community evolution. BlogPulse [3] and BlogWatcher [4] give the trend graphs which shows the popularity of specific topics over time. Adar et al. check and visualize how information is tracked in blog network with similarity of texts and referred URLs, timing of infection and so on [5,6].

Regarding the ranking algorithms of pages, on the other hand, hyperlinks have been used for measures generally. PageRank [7] is calculating ranking of Web pages for a specific topic based on the idea: the page linked from many pages is important, and the page linked from many important pages is more important. HITS algorithm [8] grades the score for pages with similar way using the concepts of hub (the page which have many links for authorities) and authority (the page linked from many hubs). Some research on blogs are inspired on these lines of studies. For example, EigenRumor algorithm [9] ranks blogs with the contribution which is calculated by activity of entry submission and whether the entry is linked by other blogs.

Most of these works use publicly available data on the Web. It means users' behavioral data such as browsing, commenting, and browsing is sometimes difficult or impossible to obtain. So we can analyze the different page of value for every users. Though our work is limited to the blogsphere in Doblog in Japan, the detailed analysis would enhance the Weblog usage analysis.

[1] ©NTT Data Corp., ©hottolink,Inc.,
http://www.doblog.com/ , using data of Oct. 2003 – Aug. 2005

3 Approach

What causes users to visit other persons' blogs? What relation most affects daily browsing behavior? If we can create a model and predict whether a user will like one blog or not, we can build a recommendation list of blogs for each user. We aim for learning which relations among two blogs are influential to users' browsing behavior, so we check the relations existing between a visitor's blog and the visited blogs.

3.1 Users' Browsing Behavior

We prepare two activities, *Visiting* behavior and *Regular Reading* behavior, for the targets of learning. By analyzing these behavior, we can guess the blogs that users tend to browse with a interest and suggest to a user that he should visit the blogs he has not been to.

- Visiting: If user U_A has accessed blog B, user U_B's blog, more than once, we call it Visiting behavior from A to B.
- Regular Reading: If user U_A has accessed blog B, more than a proportion of once a week, we call it Regular Reading behavior from A to B.

3.2 Relations Among Two Blogs

Two blogs might have many relations, but we focus indirect relations. Because we analyze the effects of relation on user's browsing behaviors, two blogs with direct relation is obvious: They are always in Visiting behavior and in high possibility in Regular Reading behavior. Then because it seems that the most influential indirect relation to users' activity is a two-hop relation, a two-hop relation is used for our analysis.

Regarding the one-hop paths between blogs, we use four relations: *Bookmark*[2], *Comment*, *Trackback* and *Regular Reading*. A measure of every 16 ($= 4 \times 4$) two-hop relations are represented by the numbers of paths that connect two blogs in two hops with the same direction. For example, if there are such relations among blog A (a start) and B (a goal) as Fig. 1, there are two Comment–Bookmark paths, one Trackback–Regular Reading paths, and so on.

3.3 Way of Analysis

In order to examine the influence of blog relations for users' browsing behavior, we conduct the analysis to check the following points.

1. *Does the number of paths have a positive or negative correlation with the browsing behavior?*

   To check the correlation, we use the browsing (Visiting/Regular Reading)

[2] A special function to Doblog. It is a link to favorite blog from one's own blog.

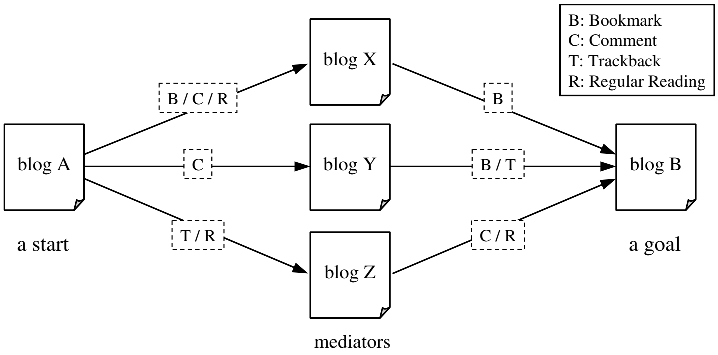


Fig. 1. Two-hop relation between blog A and B

ratio P_{browse}. If N_{browse} is the number of pairs where browsing relation is established and N_{all} is the total number of pairs, P_{browse} is defined as below:

$$P_{browse} = \frac{N_{browse}}{N_{all}}$$

Then we calculate the ratio according to the each number of paths for 16 kind of relations.

2. *Can we distinguish users' behavior by using the measure of relation?*
If the answer of previous question is *YES*, the number of paths is now valuable measure for distinct users' behavior. Then about each of two-blog pairs, a start and a goal, we checked the number of paths for 16 kind of relations and whether the browsing relation from the start to the goal is established. Using this training data, we analyzed which relation and how many numbers of the path determine the browsing activity with a machine learning algorithm C4.5[10]. The C4.5 system constructs a decision tree and we can also understand the highly influential factor to the browsing behavior.

4 Results and Studies

This section shows the results of our analysis. The analysis are conducted with using the data of 2,648 blogs, the 5% of all 52,976 Doblog users, which are selected at random. Particularly for the dataset of machine leaning, we use 201,236 of (start, goal) pairs which have more than one two-hop paths in 7,009,256 ($={}_{2648}P_2$) pairs.

4.1 Correlation Among Two-Hop Path and Browsing Ratio

The correlation between the number of paths and Visiting activity is shown in Fig. 2. This figure shows that if the number of paths increases, the Visiting ratio rises. That is same for the correlation with Regular Reading though we

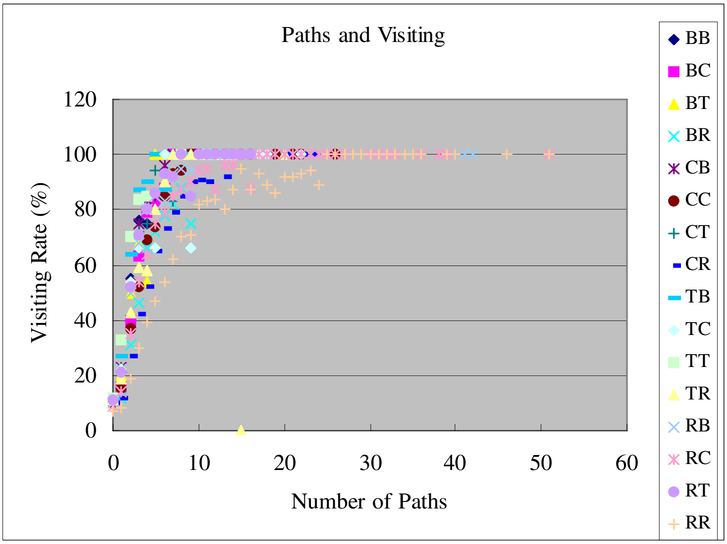


Fig. 2. Correlation of the number of paths and the Visiting rate. The characters of B, C, T, and R represent the path of Bookmark, Comment, Trackback, and Regular Reading. The first/second character in the table each represents the path of first/second hop.

Table 1. Recall and precision

behavior \ performance	recall(%)	precision(%)	F-measure(%)
Visiting	16.5	64.2	26.3
Regular Reading	11.5	54.5	18.9

omit showing the figure. The result says the number of paths have an effect to the browsing behavior and to use it for our analysis is meaningful. Next section shows the decision trees by machine learning and performance of behavior prediction.

4.2 User Behavior Prediction

Fig. 3 is a part of the decision tree for users' Visiting/Regular Reading behavior that shows which factors are effective. In these decision trees the highly influential factors occupy the upper position as nodes. The first and second characters in each node represent the path of first and second hop, the number below is the number of paths. The left/right branch diverges when there is less/more than the number of paths shown by above node. The leaves appear when it is classified to some extent. *true* and *false* represent whether the Visiting and Regular Reading relations are established or not; the value in a bracket shows the ratio of correctly classified data and below represents the ratio of data that comes this condition.

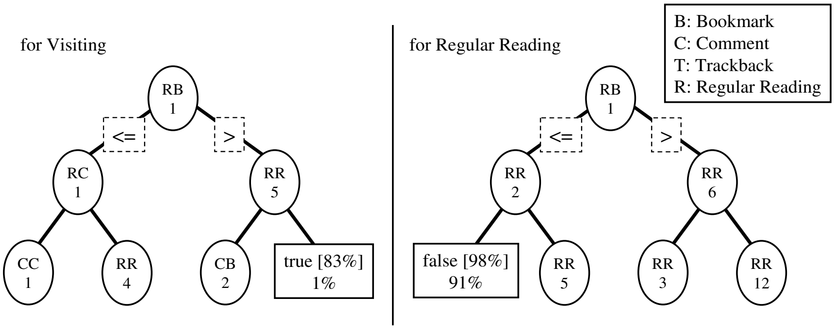


Fig. 3. Left: Top three level of decision tree for Visiting (the left) and Regular Reading (the right)

The most influential relation to both activities is the number of paths of Regular Reading–Bookmark two-hop relation. As the trees show the relations those of the first hop is Regular Reading are following, the effect from the blogs which a user often accesses is important. Particularly in the Regular Reading analysis the R–R relation occupies the most of top classes, so the relation which have a large number of paths of two hops of Regular Reading relation seem to become a direct Regular Reading relation, and the more numerous the paths, the higher the probability.

Performance of the prediction method is shown in Table 1. The recall is not high, implying that it is difficult to infer all the blogs that a user visits or regularly reads because there might be numerous reasons to read blogs. However, the precision is high: if a user is in a certain (two-hop) relation to a blog, a user is likely to visit or regularly read it. Therefore, we can reasonably recommend blogs that are predicted to have a visit or regular read relation, but which have not yet been actually visited or regularly read by the user.

5 Discussion

One of the problem of our method is the low recall ratio of prediction. The reason seems to be that users' activities can not be measured only by using two-hop relations. So we try to conduct the analysis with same way by adding the factor of *the number of paths of three-hop relation* for improvement of recall. Three-hop relation and its number of paths are defined same as two-hop relation. We can analyze that in a larger range from a start-blog than the two-hop analysis by adding this property, which is expected to make better the recall. However, the result is not very different from the two-hop analysis. In the decision trees, the factors of high influence are scarcely different from those of two-hops, and three-hop relations are only located at lower layers. While three-hop relations raise the recall a little, it down the precision awfully.

As a result, it seems that three-hop relation could not become a important property and the supposition *the most powerful indirect relation is two-hop* *relation* is right. We can advise a user to access the one-hop blogs from the ones which he regularly reads. The result – *the relation on blog networks can be the effective factor for users activity* – agree with the concept of the Semantic Web. On the other hand, prediction only from relations among blogs might be difficult. This might be one of a consideration for what kind of metadata to append to for constructing the Semantic Web.

6 Conclusion

In this study, we analyze the effect of blog network to users for their browsing activities by machine learning. We defined the measure of the unique relations in a blog by its number of paths and elucidated the influence of those relations on the users' browsing activities. Consequently, it appears that users often access the blogs in two hops from his own blog. Users especially tend to visit and be interested in the blogs in one-hop relations from his favorite blogs in high probability. This is similar to the concept of trust and we consider that such data can form the basis of information recommendation. However, we can not extract the blogs that users may read regularly at most. It seems that we should propose the other factors for analysis in the future.

References

1. Golbeck, J., Hendler, J.: Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-based Social Networks. The 14th International Conference on Knowledge Engineering and Knowledge Management (2004)
2. Kumar, R., Novak, J., Raghavan, P., Tomkins, A.: On the Bursty Evolution of Blogspace. The 12th International World Wide Web Conference (2003)
3. Glance, N., Hurst, M., Tomokiyo, T.: BlogPulse: Automated Trend Discovery for Weblogs. Workshop on the Weblogging Ecosystem, The 13th International World Wide Web Conference (2004)
4. Nanno, T., Suzuki, Y., Fujiki, T., Okumura, M.: Automatic Collection and Monitoring of Japanese Weblogs. The 13th International World Wide Web Conference (2004)
5. Adar, E., Adamic, L.A.: Tracking Information Epidemic in Blogspace. Web Intelligence 2005 (2005)
6. Adar, E., Zhang, L., Adamic, L.A., Lukose, R.M.: Implicit Structure and the Dynamics of Blogspace. The 13th International World Wide Web Conference (2004)
7. Page, L., Brin, S., Motwani, R., Winograd, T.: The Pagerank Citation Ranking: Bringing Order to the Web. Technical report, Stanford University (1999)
8. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM **46** (1999) 604–632
9. Fujimura, K., Inoue, T., Sugisaki, M.: The EigenRumor Algorithm for Ranking Blogs. The 14th International World Wide Web Conference (2005)
10. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)

UniRSS: A New RSS Framework Supporting Dynamic Plug-In of RSS Extension Modules

Eui-Hyun Jung

Dept. of Digital Media, Anyang University,
708-113, Anyang 5-dong, Manan-Gu, Anyang City, Kyunggi-do, 430-714, Korea
jung@anyang.ac.kr

Abstract. Due to the proliferation of information exchange via Internet, users suffer from information overload. For this reason, RSS is now widely adopted to deliver latest information to users without human intervention. It is also used to deliver customized data using extension modules. However an interoperability issue among RSS applications using different extension modules has been raised because the processing of extension module has to be performed in the source code level of RSS applications. To resolve this interoperability issue, we propose a new RSS framework, UniRSS, which can support extension modules via a unified interface. UniRSS suggests an architecture composed of a pair of describing schema and a delegation code model to support any kind of extension modules in the code level. It supports both RSS 1.0 and RSS 2.0, and it also provides intelligent syndication using reasoning code insertion for RSS 1.0.

Keywords: RSS, Intelligent Syndication, Semantic Web.

1 Introduction

With the explosive growth of information available at our fingertips, most Internet users experience information overload everyday. Due to information overload, users are not only overwhelmed by huge amount of data, but also spend their time catching hot information dispersed over web sites. To handle the overwhelming data, search engines were developed and showed most effective performance in their role. However, catching latest concerned information takes users enormous time and effort still yet. Users have to visit Web sites periodically for catching concerned information because of "pulling" characteristic of the Internet. To solve this issue, several frameworks using push technology have been proposed such as PointCast [1] or Marimba [2], but they were quickly dismissed because of their new but awkward and inconvenient features.

RSS (RDF Site Summary or Really Simple Syndication) proposed by W3C uses pull model but provides virtual push service to both users and content providers [3]. It allows content providers to easily post their contents with simple XML documents. Users who want to use RSS service just start a RSS reader, and then the RSS reader periodically checks changes of sites and informs up-to-date information to users.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 169 – 178, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Because of its simple and effective approach, RSS is now being used in literally thousands of sites and it is regarded as an efficient scheme for information syndication. Currently RSS has three sub versions; RDF based RSS 1.0 [4], RSS 2.0 [5] and ATOM [6] proposed by IETF. RSS 1.0 has an advantage that it can be combined with the Semantic Web. RSS 2.0 is widely adopted in Web community due to its simple structure.

Main function of RSS is information syndication, but it is also excellent in the view of extensibility to deliver customized information. RSS 1.0 can contain a RDF document called as a module and RSS 2.0 can deliver extended information to the exiting RSS channel by only adding XML schema. Due to these characteristics of RSS, applications such as PodCasting [7] and CMLRSS [8] were announced. In the case of PodCasting, when iTunesRSS [9] extension tags containing information about multimedia file are put into a RSS feed file, iTune [10] software automatically figures out the feed file and delivers additional information to users. CMLRSS is used to deliver chemistry data encoded as CML to researchers of chemistry community through RSS channel.

Although existing researches opened new application fields, an interoperability issue became raised. Most researches share RSS technology, but they have designed extension modules to be understandable only by their own proprietary applications. A design of a RSS extension module is easy for even XML beginners, but the processing of the extension module has to be implemented in the source code level of each RSS reader. Therefore, even well-known extension modules are not compatible among most RSS readers and an extension module from an individual user is also nearly neglected. Currently most RSS readers and applications are focusing on user convenience, speed and capacity but there is no concern for supporting RSS extension module dynamically. Since RSS extension module is implemented in XML, it can be used easily with any kind of application. However, corresponding processing code cannot be easily combined with other applications because it should be programmed in the part of the applications. Moreover, for RSS 1.0, this interoperability issue becomes more complicated because reasoning of RDF data in extension module can be different in each RSS application.

To solve this issue, we propose a new RSS framework, UniRSS, which can process RSS extension module in a unified way. UniRSS provides an XML schema for RSS extension module and a delegation code structure for RSS applications. The XML schema is used to describe extension module processing and the delegation code structure supports dynamic plugging of external code components that process extension modules. By adopting this method, RSS readers and applications can process RSS channels containing different extension modules without interoperability problem. UniRSS also supports both RSS 1.0 and RSS 2.0. For RSS 1.0, reasoning code for semantic data can be inserted per channel.

The remainder of this paper consists of four subsections. Issues in RSS extension and the structure of UniRSS framework for solving the issues are described in section 2. Section 3 shows reasoning code support of UniRSS for RSS 1.0. Section 4 describes implementation and evaluation of UniRSS. Finally, section 5 concludes the paper.

2 Design of UniRSS

2.1 Issues in RSS Extension

To use a RSS extension module, as shown in Fig. 1, a developer has to define an XML schema for the extension module and write additional information into the RSS feed file according to corresponding XML schema. This extended RSS feed file can be processed with a proprietary reader understanding the extension module. The problem is that other readers except the proprietary reader cannot process the extension module properly as shown in Fig.1. For example, iTunesRSS extension module distributed by Apple can even deliver rich multimedia data to users, but only few readers are able to handle the extended information properly. More seriously, this interoperability issue will quickly grow because a lot of extension modules may be released whenever an extension requirement arises.

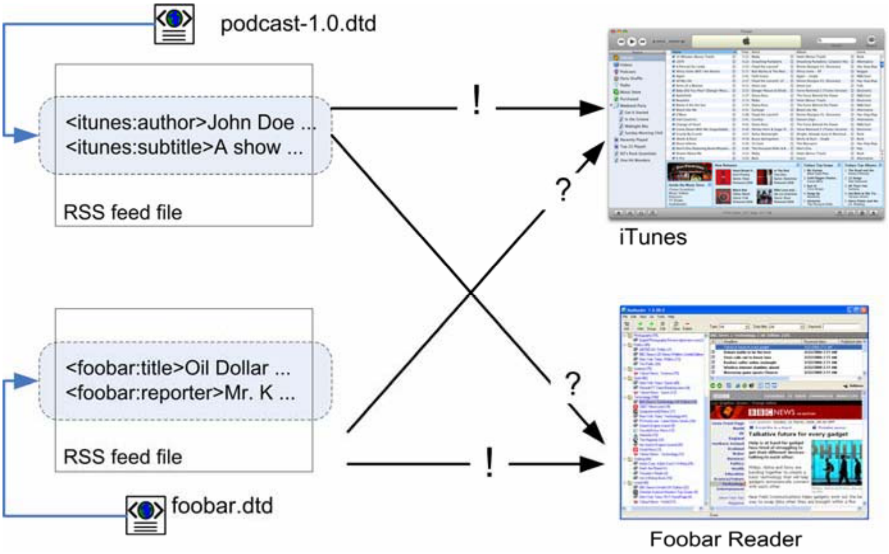

podcast-1.0.dtd
<itunes:author>John Doe ...
<itunes:subtitle>A show ...
RSS feed file
<foobar:title>Oil Dollar ...
<foobar:reporter>Mr. K ...
RSS feed file
foobar.dtd
!
?
?
!
iTunes
Foobar Reader

Fig. 1. Usage of RSS extension modules can cause an interoperability problem

This interoperability issue is caused from the difficulty of processing extension modules in RSS readers. Generally, it is easy to write a RSS extension module in a feed file. However, RSS readers have to deal with information in the RSS extension module in the source code level to show the information to user properly. Even if XML parsing in RSS extension module is easy, RSS readers have no idea how to process information contained in the RSS module. Under this situation, RSS readers must update their code to support the module whenever a new RSS extension module is released. However, this is not easy at all in the view of software development process.

This problem becomes more serious in the case of RSS 1.0. RSS 2.0 is used to deliver extended but simple data, but RSS 1.0 can contain semantic data described in RDF. The semantic data can provide flexibility but writers of RSS feed files can not determine how this data is used in advance because semantic data in RSS feed file can be understood differently by each reader. Therefore, it is more difficult to guarantee interoperability in the case of RSS 1.0 than that of using RSS 2.0. Due to these issues, the usage of RSS extension modules is limited only in a closed and proprietary community.

2.2 Detailed Structure of UniRSS

UniRSS consists of three elements as shown in Fig. 2. Fist element is an XML schema used to indicate RSS extension module. The XML Tags in the schema indicate how to process RSS extension module by RSS readers. Second element is a code plug-in interface that enables RSS readers to delegate processing of the extension module to external handler code components. This interface has a form of plug-in structure adopting Strategy design pattern [11] for using different handlers through a unified interface. Lastly, a handler is a code component that takes charge of processing the RSS extension module.

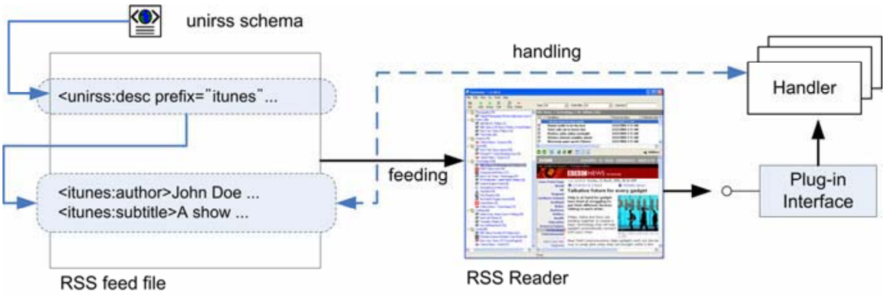

unirss schema
handling
<unirss:desc prefix="itunes"...
Handler
feeding
<itunes:author>John Doe ...
<itunes:subtitle>A show ...
Plug-in Interface
RSS feed file
RSS Reader

Fig. 2. Elements of UniRSS

A RSS extension module provider who wants to use the UniRSS framework should make a handler that processes his/her own RSS extension module. The UniRSS framework guarantees interoperability by delegating processing of extension module to the external handler written by providers.

XML Schema for Describing RSS Extension Module. XML schema for describing RSS extension module proposed in UniRSS is as follows.

```
<xs:schema targetNamespace="http://www.unirss.net/2005/unirss"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="desc">
        <xs:complexType>
            <xs:attribute name="prefix" type="xs:string" use="required"/>
            <xs:attribute name="handler" type="xs:string" use="required"/>
            <xs:attribute name="schema" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Only "desc" element exists in UniRSS schema and the element has "prefix", "handler" and "schema" attributes. The "prefix" attribute is used to indicate extension module in a feed file. The "schema" attribute contains an XML schema file name of RSS 2.0, or an ontology file name of RSS 1.0. Lastly, the "handler" attribute is the name of component containing handler code for processing of the extension module. A sample feed file containing iTunes extension module and UniRSS tag is as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<rss xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd"
xmlns:unirss="http://www.unirss.net/2005/unirss-1.0.xsd" version="2.0">
 <unirss:desc prefix="itunes" schema="podcast-1.0.dtd" handler="itune_handler"/>
 ….
<itunes:author>John Doe</itunes:author>
….
```

Code Plug-In Interface and Implementation. After a RSS reader gets a feed file containing extension module, the reader should be able to call an appropriate handler code component for the extension module in a unified way. This paper uses the Strategy design pattern to support this plug-in code structure. Strategy design pattern is one of the design patterns in GoF [11] and can connect several algorithms via the same interface. To adopt Strategy design pattern, all the handler components used in UniRSS must implement the following interface.

```
public interface RSSHandler {
    public void procTag(String tagName, Object tagValue, Hashtable attrTable);
    public Hashtable dispExtension();
    public boolean isShown();        // for RSS 1.0 only
    public void fireReasoning();  // for RSS 1.0 only
}
```

Whenever a RSS reader encounters an XML tag containing the prefix appointed by UniRSS tag, it calls procTag() of corresponding handler. Then, this handler takes tag name, tag value, and group of attributes as parameters shown in Fig. 3.

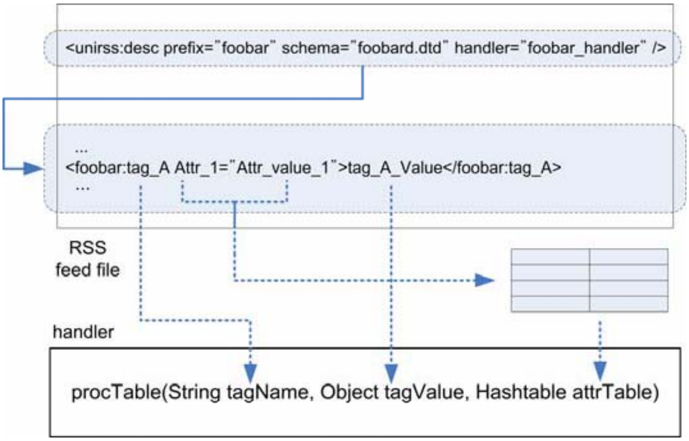
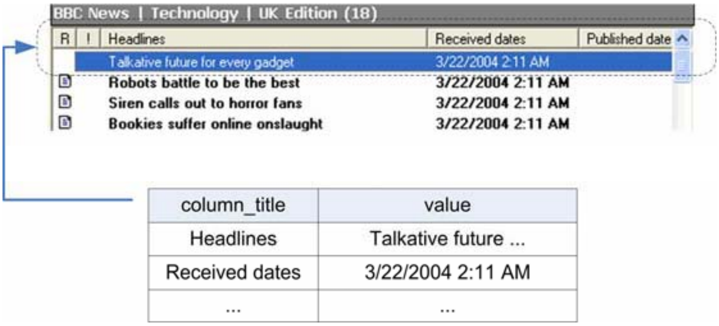

<unirss:desc prefix="foobar" schema="foobard.dtd" handler="foobar_handler" />
...
<foobar:tag_A Attr_1="Attr_value_1">tag_A_Value</foobar:tag_A>
...
RSS feed file
handler
procTable(String tagName, Object tagValue, Hashtable attrTable)

Fig. 3. A Mapping between a RSS feed file and the handler's function

Handler code component processes given parameters and stores results in the hash table as a form of [column_title, value] pair. Most RSS readers show feed data as a table format as shown in Fig. 4. Therefore, the result value of extension module processing should be returned as a table. Since column titles and values can be varied per each extension module, the result value should be returned in the form of [column_title, value].



BBC News | Technology | UK Edition (18)

R	!	Headlines	Received dates	Published date
		Talkative future for every gadget	3/22/2004 2:11 AM	
		Robots battle to be the best	3/22/2004 2:11 AM	
		Siren calls out to horror fans	3/22/2004 2:11 AM	
		Bookies suffer online onslaught	3/22/2004 2:11 AM	

column_title	value
Headlines	Talkative future ...
Received dates	3/22/2004 2:11 AM
...	...

Fig. 4. Relation between data structure and data display in a feed reader

After the feed processing is over, a RSS reader calls dispExtension() method of corresponding handler. The handler returns the hash table that contains [column_title, value] to be displayed on the RSS reader user interface.

3 Customizing Reasoning Support for RSS 1.0

The main difference between RSS 1.0 and RSS 2.0 is that RSS 1.0 is based on RDF. Since RSS 1.0 is able to contain RDF document, RSS feed files using RSS 1.0 are able to contain semantic data using several ontologies and they can be use for intelligent syndication. For example, if FOAF [12] ontology is used in a RSS feed file, a RSS reader notifies users when it finds an article uploaded by its user's friends. However, to get this kind of reasoning function, it also has to be supported in the source code level.

Since a RSS feed file writer cannot enforce RSS readers to use specific reasoning functions, it is more difficult to guarantee interoperability for RSS 1.0 than RSS 2.0. To provide intelligent syndication function and interoperability together in RSS 1.0, it is desired to delegate processing of extension module to the external code component like RSS 2.0 and to let each RSS reader have its own reasoning function. UniRSS is designed to be able to put Jess code for reasoning function when setting up RSS 1.0 feed channels. In UniRSS, a handler cooperates with Jess code for the given feed channel and the Jess [13] code is executed in the Jess engine as shown in Fig. 5.

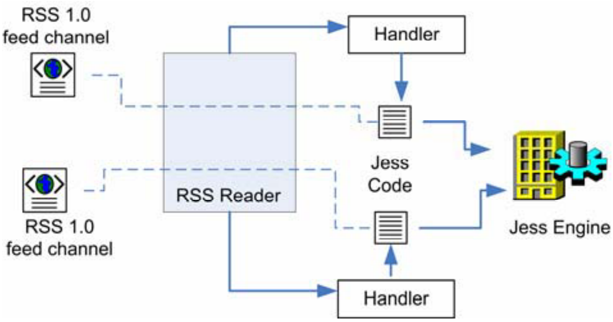


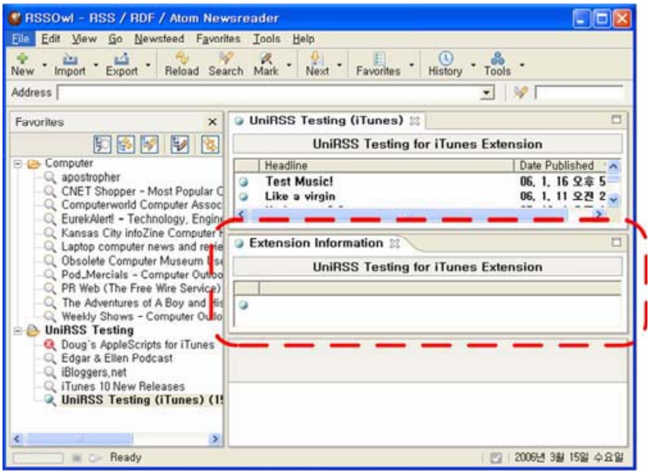
Fig. 5. Cooperation of a handler and Jess codes in the case of RSS 1.0

Like RSS 2.0, a RSS reader just calls a corresponding handler, but the handler uses Jess code via executing fireReasoning() method in RSSHandler interface. Result of the reasoning from Jess code can be linked to a call of another code or an action processing, but currently UniRSS is designed to provide intelligent syndication only. If the result is considered as negligible information by the reasoning, the return value of isShown() method is set to false and information will not be shown to the user.

4 Implementation and Evaluation

4.1 A Test Reader Supporting UniRSS

To evaluate function of UniRSS framework, a RSS test reader is implemented using open-source RSS reader, RSSOwl [14]. The user interface of implemented UniRSS


RSSOwl - RSS / RDF / Atom Newsreader
File Edit View Go Newsfeed Favorites Tools Help
New Import Export Reload Search Mark Next Favorites History Tools
Address
Favorites
Computer
apostropher
CNET Shopper - Most Popular C
Computerworld Computer Assoc
EurekAlert! - Technology, Engin
Kansas City infoZine Computer
Laptop computer news and revie
Obsolete Computer Museum Use
Pod_Mercials - Computer Outlo
PR Web (The Free Wire Service)
The Adventures of A Boy and His
Weekly Shows - Computer Outlo
UniRSS Testing
Doug's AppleScripts for iTunes
Edgar & Ellen Podcast
iBloggers.net
iTunes 10 New Releases
UniRSS Testing (iTunes) (1!
UniRSS Testing (iTunes)
Extension Information
Ready
2006년 3월 15일 수요일

UniRSS Testing for iTunes Extension

Headline	Date Published
Test Music!	06. 1. 16 오후 5
Like a virgin	06. 1. 11 오전 2

UniRSS Testing for iTunes Extension

Fig. 6. Snapshot of a test reader supporting UniRSS

test reader is shown in Fig. 6. Unlike other readers, the test reader has a tab that shows extension information. When a user selects a RSS feed containing extension module, the extension information will be displayed in the tab.

Each handler component is located in the handler folder as jar format for implemented reader to be able to process extension modules. The jar file must contain a handler code component as form of java class file. In the jar file, a schema file will be contained for RSS 2.0 or an ontology file for RSS 1.0. Especially for RSS 1.0, Jess code can be inserted per channel.

4.2 RSS 2.0 Feeding

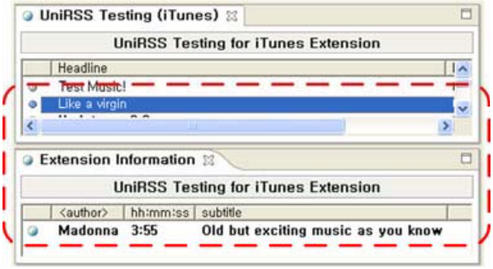
To test valid operation of RSS 2.0 extension module in the UniRSS, RSS feed file including iTunesRSS is set as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<rss xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd"
xmlns:unirss=”http://www.unirss.net/2005/unirss-1.0.xsd” version="2.0">
<unirss:desc prefix=”itunes” schema=”podcast-1.0.dtd”handler=”itune_handler”/>
<item>
<title>Like a virgin</title>
<itunes:author>Madonna</itunes:author>
<itunes:duration>3:55</itunes:duration>
<pubDate>Mon, 16, Jan 2006. 17:00:00 GMT</pubDate>
<itunes:subtitle>Old but exciting music as you know</itunes:subtitle>
</item>
```

iTunesRSS handler's code is programmed as follows and we checked it in UniRSS reader.

```
public class ItuneHandler {
        Hashtable tbResult;
        public void procTag(String tagName, Object tagValue, Hashtable attrTable)
        {
                    if(tagName.equals("author")) {
                                tbResult.put("<"+tagName+">", tagValue);
                    } else if(tagName.equals("duration")) {
                                tbResult.put("hh:mm:ss", tagValue);

                    } else {
                                tbResult.put(tagName, tagValue);
                    }
        }
}
```

After locating feed file on the test server, the test UniRSS reader displays extension information of corresponding feed file in the extension tab shown in Fig. 7. As shown in Fig. 7, the data in iTunes module such as an author or duration is processed by the handler code.



UniRSS Testing (iTunes)

UniRSS Testing for iTunes Extension

Headline
Test Music!
Like a virgin

Extension Information

UniRSS Testing for iTunes Extension

<author>	hh:mm:ss	subtitle
Madonna	3:55	Old but exciting music as you know

Fig. 7. Results of iTunes module processing

4.3 RSS 1.0 Feeding

For RSS 1.0, a reasoning code can be attached in addition to simply displaying extension information. UniRSS framework allows users to add Jess code per channel that enables customized reasoning. For a test, FOAF ontology is included in a RSS feed file as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns="http://purl.org/rss/1.0/"
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:foaf="http://xmlns.com/foaf/0.1"
  xmlns:unirss=" http://www.unirss.net/2005/unirss/ >
 <unirss:desc prefix=”foaf” schema=”foaf.rdf” handler=”foaf_handler”/>
<item rdf:about="http://xml.com/pub/2000/08/09/xslt/xslt.html">
    <title>Processing Inclusions with XSLT</title>
    <link>http://xml.com/pub/2000/08/09/xslt/xslt.html</link>
    <foaf:name>John</foaf:name>
    <description> sample description.</description>
</item>
```

Code shown below is put into the Jess code of corresponding feed channel of UniRSS reader. The given Jess code is used to display an article only when the article publisher is a friend of the user.

```
(deftemplate friend (slot name) (slot email))

(deffacts add-data
    (friend (name "John") (email "john@anyang.ac.kr"))
    (friend (name "Bred") (email "bred@foobar.com"))
)

; Finds friends of mine and returns a pair of {"E-mail","friend's email"}
(defrule is-friend
    (friend (name ARG_1) (email ?em))
    =>
    (store COL_TITLE "E-mail")
    (store COL_VALUE ?em)
)
```

When feed files containing publishers as John, Bred and Jane exist on different channels, only those RSS feeds whose publishers are John and Bred are shown in the test RSS reader. This result shows that RDF document contained in RSS 1.0 can be processed based on the user's reasoning code.

5 Conclusion

RSS is considered as an effective syndication technology that informs users of up-to-date information using simple XML and HTTP. Moreover, RSS is widely used for making customized syndication applications with an extension module and a lot of extension modules have been proposed. However, interoperability among applications using different extension modules is not guaranteed because the interoperability depends on how modules are processed in each RSS applications in the source code level. This situation results in interoperability problem from which extension modules tend to be only utilized within proprietary domains.

In this paper, UniRSS framework for supporting RSS extension modules dynamically is proposed. UniRSS suggests an XML schema indicating extension modules to be processed and a unified plug-in code interface. In UniRSS framework, extension modules are not only indicated with the proposed schema, but also processed by the handler in the delegation mechanism. Using this approach, any RSS extension module can be supported by RSS applications with a unified way. UniRSS also provides an architecture that enables intelligent syndication by allowing dynamic inserting of Jess code per channel. It allows RSS applications to have reasoning function of semantic data in RSS 1.0. Proposed architecture using a pair of describing schema and its corresponding handler can be a new approach for Semantic Web interoperability where a variety of ontologies are mingled together.

References

1. Satish, R., Vihba, D.: The PointCast Network. Proc. of ACM SIGMOD (1998) 520
2. Marimba Inc.: Technical White Paper-Introducing Castanet. (1999)
3. Hammersley, B.: Content Syndication with RSS. O'Reilly & Assoc. (2003)
4. Gabe, B. and et.al: RDF Site Summary (RSS 1.0). http://web.resource.org/rss/1.0/spec (2001)
5. Winer, D.: RSS 2.0 Specification. Berkman Center for Internet & Society at Harvard Law School. (2005)
6. Mark, N., Robert, S.: ATOM Syndication Format. RFC 4287 (2005)
7. Wikipedia: Podcasting from Wikipedia. http://en.wikipedia.org/wiki/Podcasting
8. Murray-Rust, P. and et.al.: Chemical Markup, XML and the World Wide Web. Part 5. Applications of Chemical Metadata in RSS Aggregators. Journal of Chemical Information and Computer Sciences, Vol. 44, No. 22. (2004) 462-469
9. Erich, G., Richard, H., Ralph, J., John, V.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley (1995)
10. Wikipedia: iTunes from Wikipedia. http://en.wikipedia.org/wiki/Podcasting
11. Apple Inc.: Podcasting and iTunes: Technical Specification. http://www.apple.com/itunes/podcasts/techspecs.html
12. Dan, B., Libby, M.: FOAF Vocabulary Specification. http://xmlns.com/foaf/0.1/. (2005)
13. Ernest. F.: Jess in Action. Manning (2003)
14. RSSOwl: RSSOwl – A Java RSS/RDF/Atom News Reader. http://www.rssowl.org

Ontology-Based RBAC Specification for Interoperation in Distributed Environment

Di Wu[1], Xiyuan Chen[2], Jian Lin[2], and Miaoliang Zhu[3]

[1] College of Computer Science, Zhejiang University,
310027, Hangzhou, China
wudi@zju.edu.cn
[2] College of Computer Science, Zhejiang University,
310027, Hangzhou, China
{cxyspirit, appolin}@zju.edu.cn
[3] College of Computer Science, Zhejiang University,
310027, Hangzhou, China
zhum@zju.edu.cn

Abstract. Today, the formulation, specification, and verification of adequate data protection policies in open distributed environment appear as the main challenge to address concerning authorization. Role-based access control models have attracted considerable research interest in recent years due to their innate ability to model organizational structure and their potential to reduce administrative overheads. This paper proposes ontology specification to describe Role-based Access Control model and extend it with a general context expression. Based on these definitions, the specification for interoperation in distributed environment is introduced. The works include a definition of ontology to describe the concepts and a declaration of rules to explicit the relationship between concepts. The ontology based approach can express security policy with semantic information and provide a machine interpretation for descriptions of policy in open distributed environment.

Keywords: Ontology, RBAC, access control policy, interoperation.

1 Introduction

With the rapid development of information technologies, there is a growing concern for security and privacy from traditional stand-alone system to open distributed environment. In open, heterogeneous, distributed environment there is a great likelihood that inconsistent interpretations will be made of the security information in different domains. It intensifies the need for robust access control management to share and integrate security policy in such environment.

In the context of the Semantic Web, ontology provides formal specification of concepts and their interrelationships, and has the advantage of dealing with interoperation at semantics level over heterogeneous environments. The key challenge for ontology based policy specification of access control is to define related concepts and relationships and develop uniform representation by ontology.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 179 – 190, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Role-based access control (RBAC) models have generated great interest in the security community as a powerful and generalized approach to security management [1]. RBAC models show clear advantages over traditional discretionary and mandatory access-control models [2]. In this paper, we propose Ontology Role-based Access Control (O-RBAC) specification using semantic language-OWL (Web Ontology Language) [3] and rule language-SWRL (Semantic Web Rule Language) [4]. The O-RBAC extends core RBAC [5] model with a general expression of context to enrich description for RBAC model. Moreover, the ontology and rules for interoperation in distributed environment are defined based on core RBAC and extended context specification.

The rest of the paper is organized as follows. In section 2, some related works are introduced. The ontology and rules specification of core RBAC are proposed in section 3. The ontology-based context specification for extension of RBAC is presented in section 4. And section 5 introduces the definition of ontology and rules for interoperation in distributed environment. Last a conclusion is made and prospects of the future work are looked forward.

2 Related Works

Recently, a lot of researches appear in the area of using XML to express Role-Based Access Control (RBAC) policy, such as XACML (eXtensible Access Control Markup Language) [6], X-RBAC (XML Role-Based Access Control) [7].

XACML is an OASIS standard specification now. It defines a general policy language based on XML used to express access control policies and protect resources as well as an access decision language. It can also combine multiple rules and multiple policies under various modes (deny - overrides, permit-overrides) and can be configured to support role-based access control and usage-oriented resource protection policies.

X-RBAC is based on an extension of the role-based access control model and it provides a framework for specifying mediation policies in a multi-domain environment and allows specification of RBAC policies and extends RBAC model with temporal constraints, role attributes, contextual conditions, a notion of role states, and preconditions for state transitions.

Both XACML and X-RBAC are based on XML, and they have applied to many applications in distributed environment. However, XML comes short on providing a machine interpretation of policy specification; XML DTDs and XML Schemas allow more flexibility in the syntactic descriptions of data, but do little to agree the interpretation of that data; each application domain has to agree on the meanings of terms. Providing a machine interpretation for descriptions of security policy specification in distributed environment is very important for avoiding security vulnerabilities associated with the misuse of a specified policy or its erroneous deployment.

Research of the Semantic Web also has focused on how to describe security requirements gradually. KAoS uses OWL as the basis for representing and reasoning about policies within Web Services, Grid Computing, and multi-agent system platforms [8], [9]. KAoS also exploits ontology for representing and reasoning about domains describing organizations of human, agent, and other computational actors. Rei is a new deontic logic-based policy language that is grounded in a semantic representation of policies in RDF-S, although the authors are moving towards an OWL implementation [10]. Ponder is an object oriented policy language for the management of distributed systems and networks [11]. The developers of Ponder pioneered many of the policy management concepts used in KAoS and Rei, though its implementation differs in important ways. But these works have not been involved in RBAC policy specification.

In order to make application in distributed environment to understand and interpret security policy correctly, we define ontology to help express RBAC based access control policy using OWL and related rules to enhance expressive and deducible ability using SWRL.

3 Ontology-Based Specification for RBAC

3.1 Ontology for RBAC

With RBAC, users cannot associate with permissions directly but roles. Permissions must be authorized for roles, and roles must be authorized for users. A user assigned to a role can activate the role in a session and acquire all the permissions assigned to it. And constraints can be applied to the assignment of users and permissions to roles and users' activation of roles in sessions.

As shown in Fig.1, the ontology for RBAC model is designed to help expressing concepts and relationships, and the formal meanings of them can be interpreted by machine. The classes *Users*, *Roles*, *Permissions* and *Sessions* are defined to express the key concepts in RBAC, and several types of property are defined to describe relationships among them.

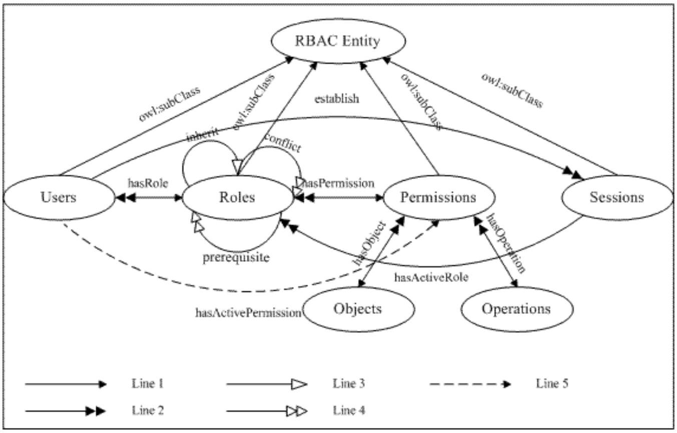


Fig. 1. Ontology for RBAC model

(1) Subclass Relationship. The class *RBAC Entity* is an abstract concept defined in O-RBAC specification and it can be used to describe some commonness of elements in RBAC. So the classes *Users*, *Roles*, *Permissions* and *Sessions* all have the property *subClassOf* related with class *RBAC Entity* and represented by line 1.

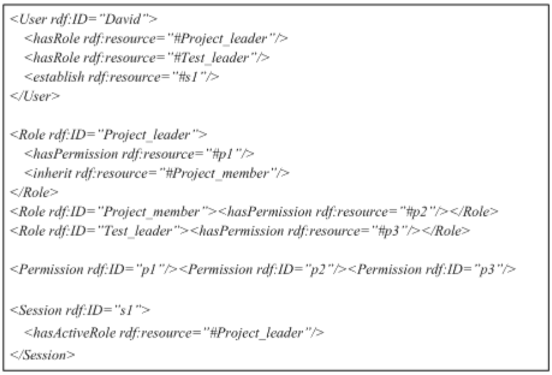
(2) Basic Relationship. Line 2 expresses the basic relationships in RBAC model. For example, properties *hasRole* and *hasPermission* respectively express assignment relationship among class *User*, *Role* and *Permission*. And properties *establish* and *hasActiveRole* show the meaning that users can establish sessions during which they may activate a subset of the roles they belong to.

(3) Hierarchies Relationship. Hierarchies are a natural means for structuring roles to reflect an organization's lines of authority and responsibility. The senior role can inherit all permissions from junior role. And the property *inherit* is denoted by line 3 to express hierarchies relationship between roles.

(4) Constraints Relationships. Constraints are an important aspect of RBAC and a powerful mechanism for laying out higher level organizational policy. Line 4 represents Constraints Relationships in the figure. For example, the property *conflict* represents the most common RBAC constraint, mutually exclusive roles, which mean the same user can be assigned to at most one role in a mutually exclusive set. And the property *prerequisite* represents the prerequisite roles constraint. It means a user can be assigned to role A only if the user already is assigned to role B.

(5) Indirect Relationship. Line 5 expresses indirect relationships among elements in RBAC. These relationships are not expressed directly in specification, but can be retrieved with help of rules and reasoning. Related rules will be introduced in next section.

Based on ontology for RBAC model, individuals and relationships can be defined according to access control requirements. And such policy specification can be exchanged among applications in distributed environment. Applications can use it to manage and maintain access control information in semantic level. Fig.2 shows an instance of ontology based access control policy and the information can be retrieved as below: the user *David* has roles *Project_leader*, *Test_leader* and can establish a session, in which only the role *Project_leader* can be activated; the role *Project_leader* can inherit all permissions from the role *Project_member*.



```
<User rdf:ID="David">
  <hasRole rdf:resource="#Project_leader"/>
  <hasRole rdf:resource="#Test_leader"/>
  <establish rdf:resource="#s1"/>
</User>

<Role rdf:ID="Project_leader">
  <hasPermission rdf:resource="#p1"/>
  <inherit rdf:resource="#Project_member"/>
</Role>
<Role rdf:ID="Project_member"><hasPermission rdf:resource="#p2"/></Role>
<Role rdf:ID="Test_leader"><hasPermission rdf:resource="#p3"/></Role>

<Permission rdf:ID="p1"/><Permission rdf:ID="p2"/><Permission rdf:ID="p3"/>

<Session rdf:ID="s1">
  <hasActiveRole rdf:resource="#Project_leader"/>
</Session>
```

Fig. 2. An example of access control policy based on ontology for RBAC model

3.2 Rules for RBAC Model

The ontology describes concepts and relationships for RBAC model through class and property. Besides the ontology, we give more explicit meaning to the properties by defining some rules. And the formal rules can be used by known reasoning algorithms and implemented systems to improve comprehension of machine [12], [13].

There are two types of rules defined in O-RBAC specification, Constraint Rule and Function Rule, listed in Table 1.

Table 1. Rules for RBAC Model

Session Constraint Rule: a user establishes a session and the session has an activated subset of the set of roles the user is assigned to.	
$establish(?u, ?s) \wedge hasRole(?u, ?r) \rightarrow hasActiveRole(?s, ?r)$	(1)
Mutually Exclusive Roles Constraint Rule	
$hasRole(?u, ?r) \wedge conflict(?r, ?r') \rightarrow \neg hasRole(?u, ?r')$	(2)
Prerequisite Roles Constraint Rule	
$hasRole(?u, ?r) \wedge prerequisite(?r', ?r) \rightarrow hasRole(?u, ?r')$	(3)
Indirect Relationship Function Rule	
$establish(?u, ?s) \wedge hasActiveRole(?s, ?r) \wedge hasPermission(?r, ?p)$ $\rightarrow hasActivePermission(?u, ?p)$	(4)
Hierarchies Relationship Function Rule	
$inherit(?r', ?r) \wedge hasPermission(?r, ?p) \rightarrow hasPermission(?r', ?p)$	(5)

- **Constraint Rule**

This type of rule is a restriction of relationship. In maintenance of access control information, before some changes happen, Constraint Rule can help applications to validate whether the relationship can be established after changes. For example, if a user has established a session and wants to activate a role in the session, we must check whether the user has this role through the rule (1) in Table 1. If the user has the role, then the related access control information can be updated in application.

- **Function Rule**

As described in previous section 3.1, there are some relationships are not expressed directly in policy specification and they can be found by reasoning with Function Rule. When users and administrator want to browse access control information in application, Function Rule can help application provide convenient and intuitionistic view to simplify operation. In Fig.3, if a user wants to find all the permission related with him, including inherited permission with role hierarchy mechanism, it's very complex to operate access control information, but with rule (5) in Table 1 it's easier to implement.

In this paper, the expressions of some properties and related rules are simplified, such as *¬hasRole(?u,?r)* in rule (2) shows the semantic that the user can not be assigned to the role. In fact, the property *cannotHasRole* needs to be defined alone and then can be used in definition of rules with SWRL.

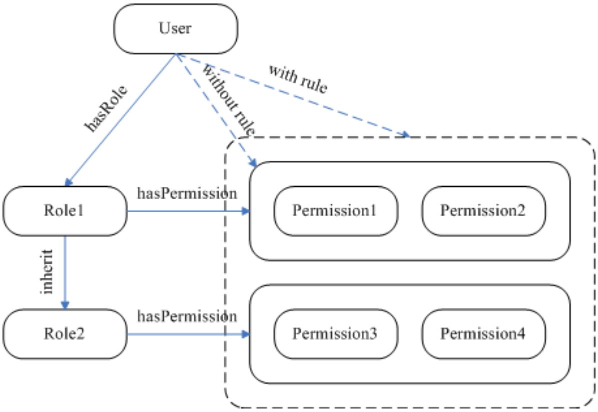


Fig. 3. The effect of Function Rule

4 Ontology-Based Context Specification for RBAC

The notion of contexts has been around a long time, and there is no consensus on the semantics of a context. Cognitively, a context consists of a set of data attributes that vary according to the context in which they are viewed. By defining proper contexts and restriction rules associated with them, it's very flexible to extend RBAC model. With the length limitation of this paper, we only pick up two types of context to denote how to define ontology-based context specification for RBAC Model by a general way.

4.1 Context Ontology

There are many discussions about extension of RBAC model with attribute, for instance, role attributes to associate different states with roles, time context to restrict the duration of session and so on [7], [14]. And all kinds of attributes can be looked as context used to describe elements in RBAC model. As shown in Fig.4 (a), class *Context* is defined to express abstract context in RBAC model and there are four basic types of contexts also defined as subclass of *Context*: *RoleContext*, *UserContext*, *SessionContext* and *PermissionContext*.

- **State Context**

Depending on the application semantics, all roles might not be available to all users at all times. So role can be associated with different states. As shown in Fig.4 (b), the state context is one type of role context, so *StateContext* is defined as the subclass of *RoleContext* and it also has three subclasses. *DisabledState* indicates that the role can't be activated in a session. *EnabledState* indicates that the users authorized for the role at the time of the request can activate the role. *ActiveState* implies that at least one user has activated the role. Application can generate events to transit role from one state to others.

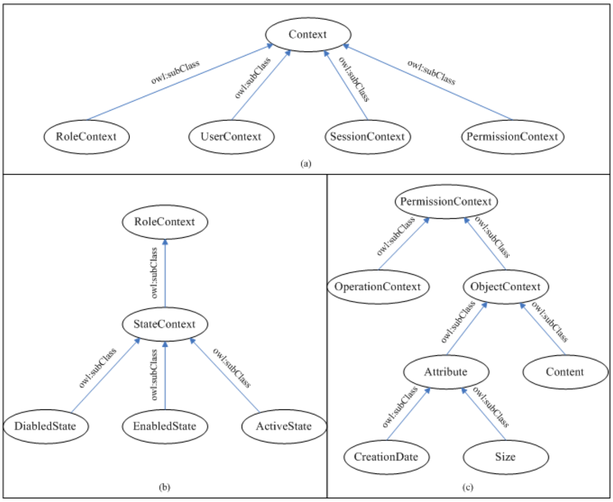


Fig. 4. (a) Basic context ontology for RBAC model (b) State context ontology for Role (c) Object Context ontology

- **Object Context**

In RBAC model, permission is composed of object and operation. And object can be based on any classifiable attribute, including its date of creation, object size, object type (image, text, streaming video, etc.), or information about the contents of the object (for example, objects can be classified based on whether they contain any content related to education, finance or insurance). So class *ObjectContext* is defined as subclass of class *PermissionContext* and also has its own subclass *Attribute* and *Content*. And context can be classified further by defining class *CreationData* and *Size* as subclass of class *Attribute*. The ontology for these contexts is described in Fig.4 (c).

4.2 Relate Context with RBAC

In Fig.5 (a), several properties can be found, such as *hasRoleContext*, *hasPermissionContext*, *hasUserContext* and *hasSessionContext*, and they are used to relate context with corresponding element in RBAC model. It seems very easy to extend RBAC specification with context. But there will be another problem, if basic contexts described in Fig.4 (a) need to be extended by defining subclass, such as *StateContext* and *ObjectContext*. For example, the state context can only express the meaning as a role context with the property *hasRoleContext*. So there is an involuntary thought to define the property *hasStateContext*, which means a role has a state context. Every time, user needs to build two relationships between the same role and state context to hold above two meanings. It's expectable that the specification for various contexts and operation on them will be complicated.

In OWL, property is used to be express relationship among concepts and the relationship can be inherited. The property hierarchies may be created by making one or more statements that a property is a subproperty of one or more other properties. And

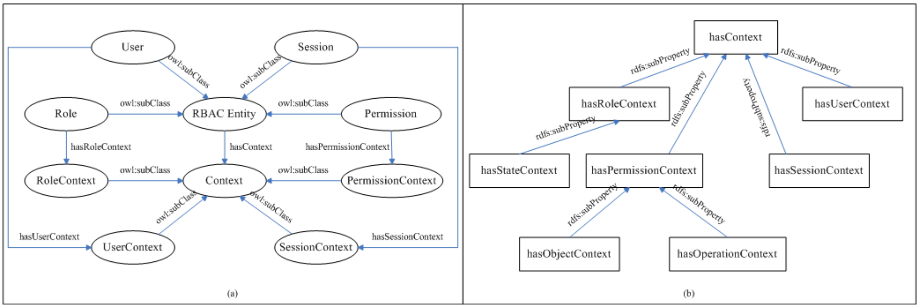


Fig. 5. (a) Ontology for RBAC with context (b) Context relationship hierarchies

subproperty can expresses further meaning of relationship. For example, *hasSibling* may be stated to be a subproperty of *hasRelative*. From this a reasoner can deduce that if an individual is related to another by the *hasSibling* property, then it is also related to the other by the *hasRelative* property. So as shown in Fig.5 (b), in addition to properties for various contexts, context hierarchies are defined with subproperty. And the property *hasStateContext* is defined as subproperty of the property *hasRoleContext*, and then two meanings can be held easily.

Based on the mechanism of property hierarchies, the context specification in O-RBAC has two advantages.

(1) Expansibility. People can customize context specification according to different access control requirements without change of O-RBAC specification.

(2) Flexibility. There is abundant semantic information that can be used in different environment. Perhaps for the same policy, only basic information of context related to a role by property *hasRoleContext* is needed by user and an access control runtime system will analyze detail of context with the meaning of property *hasStateContext*.

4.3 Rules for Context

Like introduction in section 3.2, some rules can be defined based on context specification to express more explicit meaning.

If a user has established a session and want to active a role in the session, the validation whether or not the role has *EnabledState* context must be executed first. Here the properties *hasEnabledState* and *hasDisabledState* are defined as subproperty of the property *hasRoleContext*. Then the Constraint Rules group (6) is defined to describe above meaning.

$$\begin{cases} establish(?u,?s) \wedge hasRole(?u,?r) \wedge hasEnabledState(?r,?e) \\ \rightarrow hasActiveRole(?s,?r) \\ establish(?u,?s) \wedge hasRole(?u,?r) \wedge hasDisabledState(?r,?e) \\ \rightarrow \neg hasActiveRole(?s,?r) \end{cases} \tag{6}$$

Function Rule can be defined to describe object with given conditions based on object context. The property *hasSize* is the subproperty of property *hasObjectContext* and class *ConditionObject* is the subclass of class *Object* to express result set retrieved by Function Rule. The rule (7) defines the object, the size of which is more than 10M. Here, the property *hasValue* means class *Size* has a data property *Value* with float type.

$$\begin{aligned} & hasSize(?o, ?s) \wedge hasValue(?s, ?v) \wedge swrlb: moreThan(?v, 10) \\ & \rightarrow ConditionObject(?o) \end{aligned} \tag{7}$$

5 Ontology-Based Specification for Interoperation

In distributed environment, avoiding violations occur during interoperation is the main challenge and two principles must be enforced [15]:

- *Autonomy*. An access that's permitted within an individual system must also be permitted under secure interoperation.
- *Security*. An access that isn't permitted within an individual system can't be permitted under secure interoperation.

In [7], two architectural configurations characterizing a distributed environment are proposed according to above principles: loosely coupled and federated coupled distributed environments. In a loosely coupled distributed environment, independent systems dynamically come together to share information for a period of time. In a federated distributed environment, one system is typically designated master; the others are local domains. The master mediates accesses to individual systems through a global policy.

In both environments, the mapping information is needed for interoperation as illustrated in Fig.6. The solid line describes inheritance relationship among roles. The

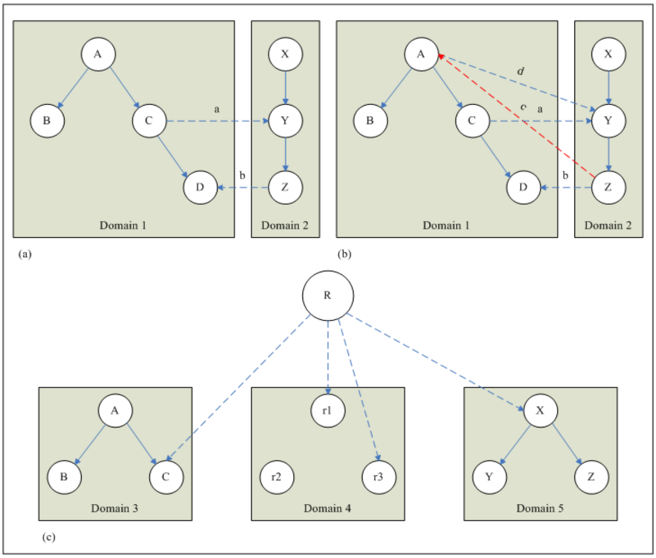
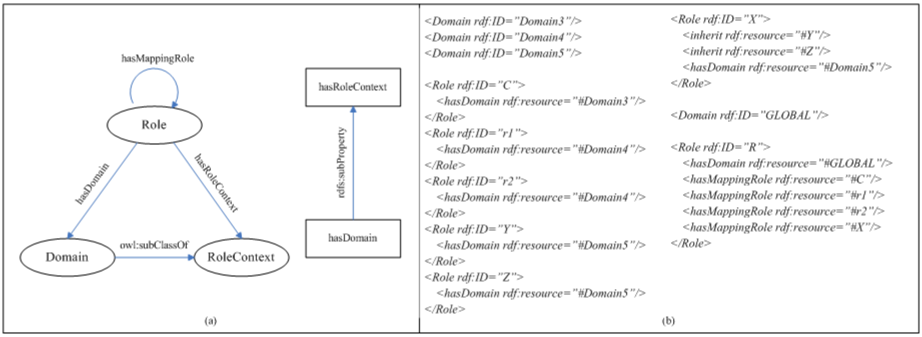


Fig. 6. (a) Interoperation in loosely coupled distributed environment (b) Violation in loosely coupled distributed environment (c) Interoperation in federated coupled distributed environment

dashed means roles mapping between different domains and the mapping relationship is directional. In Fig.6 (a), link *a* expresses interoperation between *domain 1* and *domain 2* such that users authorized for role *C* are also authorized for roles *Y* and *Z*. And Fig.6 (c) shows that roles mapping from global role *R* to local roles in the three domains and roles a user assigned to *R* may assume in the local domains in a federated distributed environment.

Ontology defined for RBAC and context is extended in Fig.7 (a) to describe the semantics for interoperation. The class *Domain* is defined as subclass of the class *RoleContext* and related with class *Role* through property *hasDomain*, the subporperty of the property *hasRoleContext*. And the property *hasMappingRole* is used to express the relationship of roles mapping.



```
<Domain rdf:ID="Domain3"/>
<Domain rdf:ID="Domain4"/>
<Domain rdf:ID="Domain5"/>

<Role rdf:ID="C">
   <hasDomain rdf:resource="#Domain3"/>
</Role>
<Role rdf:ID="r1">
   <hasDomain rdf:resource="#Domain4"/>
</Role>
<Role rdf:ID="r2">
   <hasDomain rdf:resource="#Domain4"/>
</Role>
<Role rdf:ID="Y">
   <hasDomain rdf:resource="#Domain5"/>
</Role>
<Role rdf:ID="Z">
   <hasDomain rdf:resource="#Domain5"/>
</Role>
```

```
<Role rdf:ID="X">
   <inherit rdf:resource="#Y"/>
   <inherit rdf:resource="#Z"/>
   <hasDomain rdf:resource="#Domain5"/>
</Role>

<Domain rdf:ID="GLOBAL"/>

<Role rdf:ID="R">
   <hasDomain rdf:resource="#GLOBAL"/>
   <hasMappingRole rdf:resource="#C"/>
   <hasMappingRole rdf:resource="#r1"/>
   <hasMappingRole rdf:resource="#r2"/>
   <hasMappingRole rdf:resource="#X"/>
</Role>
```

Fig. 7. (a) Ontology for interoperation (b) An example of roles mapping in federated coupled distributed environment based on ontology for interoperation

Fig.7 (b) shows an instance of ontology based access control policy described in Fig.6 (c). In addition to explicit definition of ordinary domains, a global domain is defined to contain global roles. So the definition of roles mapping, based on ontology for interoperation, in federated coupled distributed environment is consistent with loosely coupled one.

In such interoperation, the roles mapping relationship may introduce a cycle in the interoperation lattice enabling a subject lower in the access control hierarchy to assume the permissions of a subject higher in the hierarchy. Fig.6 (b) illustrates a violation of the security principle as mentioned at beginning of this section. Two links of roles mapping, *c* and *d*, are added into interoperation between *domain 1* and *domain 2*. Users originally authorized for role *Z* and not for role *Y* are now authorized for role *Y* because of the roles mapping path from *Z* to *A* to *A* to *Y*. Such security principle also can be defined as rules.

Before defining rules for security principle, rules to restrict property *hasMappingRole* must be defined first. Rules group (8) means that if two roles exist in the same domain, the mapping between them can not be established. Then rules group (9) is defined to avoid violation because of cyclic conflicts. In rules group (9), r_2 is the start point of roles mapping link and r_1 is the end point. There are three possibilities can arouse cyclic conflicts:

- Existing roles mapping between junior role of r_1 and senior role of r_2.
- Existing roles mapping between junior role of r_1 and the role r_2.
- Existing roles mapping between the role r_1 and senior role of r_2.

$$\begin{cases} hasDomain(?\,r, ?\,d) \wedge hasDomain(?\,r', ?\,d) \rightarrow \neg hasMappingRole(?\,r, ?\,r') \\ hasDomain(?\,r, ?\,d) \wedge hasDomain(?\,r', ?\,d) \rightarrow \neg hasMappingRole(?\,r', ?\,r) \end{cases} \tag{8}$$

$$\begin{cases} inherit(?\,r_1, ?\,r') \wedge inherit(?\,r'', ?\,r_2) \wedge hasMappingRole(?\,r', ?\,r'') \\ \rightarrow \neg hasMappingRole(?\,r_2, ?\,r_1) \\ inherit(?\,r_1, ?\,r') \wedge hasMappingRole(?\,r', ?\,r_2) \rightarrow \neg hasMappingRole(?\,r_2, ?\,r_1) \\ inherit(?\,r'', ?\,r_2) \wedge hasMappingRole(?\,r_1, ?\,r'') \rightarrow \neg hasMappingRole(?\,r_2, ?\,r_1) \end{cases} \tag{9}$$

The constraints in RBAC combines with roles mapping also can arouse the conflicts in distributed environment. Corresponding ontology and rules for interoperation also can be defined, but details will not be discussed in this paper.

6 Conclusion

In this paper, we have introduced an approach, Ontology Role-based Access Control (O-RBAC) specification, to express access control policies based on RBAC model with OWL and SWRL, and extended RBAC model with a general specification of context. Further more the specification for interoperation in distributed environment is defined based on definition of core RBAC and context extension. It gives more formal meaning to the both concepts and relationships in security policy. The abundant semantic information helps managing, sharing and integrating RBAC based policy flexibly in distributed environment. In conclusion, the ontology based approach of RBAC specification is the foundation to achieve semantic interoperability among the different components of access control systems in open distributed environment.

The work described in this paper focuses on the specification for security policy. It does not provide a systematic modeling approach that can be used to generate the specification from runtime application of access control, parse the specification and transform it to access control runtime system. In addition, the definition of interoperation is not completed and many features in RBAC model have not been considered. In the future, we plan to improve our idea in the following 3 aspects:

(1) We will complete O-RBAC to cover more features in RBAC model.
(2) We will construct a systematic model that can be used for transformation between ontology based specification and access control runtime system.
(3) We will integrate the systematic model into open and distributed environment, such as Enterprise Application Integration, Web Services and Grid Computing etc., to resolve the authorization problem in open and distributed environment.

References

1. Bacon, J., Moody, K., and Yao, W.: A Model of OASIS Role-Based Access Control and Its Support for Active Security. ACM Trans. Information and System Security, Vol. 5, No. 4. ACM Press, New York (2002) 492–540
2. Osborn, S.L., Sandhu, R., and Munawer, Q.: Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies. ACM Trans. Information and System Security, Vol. 3, No. 2. ACM Press, New York (2000) 85–106
3. Patel-Schneider, P.F., Hayes, P., Horrocks, I., ed.: OWL: Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation 10 February 2004. Latest version is available at http://www.w3.org/TR/owl-semantics/
4. Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B. and Dean, M.: SWRL: A semantic web rule language combining owl and ruleml. W3C Member Submission, 21 May 2004. Available at http://www.w3.org/Submission/SWRL/
5. Ferraiolo, D., et al.: The NIST Model for Role-Based Access Control: Towards a Unified Standard. ACM Trans. Information and System Security, Vol. 4, No. 3. ACM Press, New York (2001) 224–274
6. Moses, T., ed.: OASIS eXtensible Access Control Markup Language (XACML) Version 2.0. 24 July 2003. Latest version is available at http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf
7. Joshi, J.B.D.: Access-control language for multidomain environments. Internet Computing, IEEE, Vol 8, Is 6. IEEE Inc., Piscataway (2004) 40–50
8. Johnson, M., Chang, P., Jeffers, R., Bradshaw, J., et al: KAoS Semantic Policy and Domain Services: An Application of DAML to Web Services-Based Grid Architectures. Submitted to the AAMAS 03 workshop on Web Services and Agent-Based Engineering, Melbourne, Australia, July, (2003)
9. Uszok, A., Bradshaw, J., Jeffers, R., Suri, N., et al.: KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction, and Enforcement. To appear in proceedings of IEEE 4th International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), Lake Como, Italy, (2003)
10. Kagal, L., Finin, T., Johshi, A.: A Policy Language for Pervasive Computing Environment. To appear in proceedings of IEEE 4th International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), Lake Como, Italy, (2003)
11. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The Ponder Policy Specification Language. In proceedings of Workshop on Policies for Distributed Systems and Networks (POLICY 2001). Springer-Verlag, LNCS 1995, Bristol, UK, (2001)
12. Lodderstedt, T., Basin, D. A., and Doser, J.: SecureUML: A UML-Based Modeling Language for Model-Driven Security. Proceedings of the 5th International Conference on the Unified Modeling Language, Dresden, Germany (2002) 426–441
13. Indrakshi Ray, Na Li, Robert France, and Dae-Kyoo Kim: Constraints: Using UML To Visualize Role-Based Access Control Constraints. Proceedings of the ninth ACM symposium on Access control models and technologies. ACM Press, New York (2004) 115–124
14. Covington, M J., Moyer, M J., and Ahamad, M.: Generalized Role-Based Access Control for Securing Future Applications. Proceedings of the 23rd National Information Systems Security Conference (NISSC 2000) , Baltimore, MD. U.S.A., Oct. 16-19, 2000.
15. Gong, L., and Qian, X.: Computational Issues in Secure Interoperation. IEEE Trans. Software and Eng., Vol. 22, No. 1. IEEE Inc., Piscataway (1996) 43–52

Business Process Collaboration Using Semantic Interoperability: Review and Framework

Ruinan Gong[1], Qing Li[1], Ke Ning[2], Yuliu Chen[1],
and David O'Sullivan[2]

[1] Department of Automation, Tsinghua University,
Beijing, China
gongruinan@gmail.com

[2] Digital Enterprise Research Institute, National University of Ireland, Galway,
Galway, Ireland
ke.ning@deri.org

Abstract. Business process collaboration is one of the most significant factors driving today's global business development. Researches and applications such as business process modeling, workflow interoperability, web service and ambient intelligence have been involved in this area. However, a holistic understanding is missing. To clarify the requirements and build a research foundation for business process collaboration, a conceptual model is provided in this paper. Then the state of the art and the future trend of business process modeling and process interoperability are reviewed based on this model. Furthermore, inspired by the novel semantic web technologies, a semantic agent based framework to facilitate business process collaboration is given.

1 Introduction

Today's enterprises have to establish cooperating partnerships to meet the challenges of changing market and high competition, which leads to inter-organization business process collaboration. In literature, research enhancing the efficiency and effectivity of process collaboration is usually divided into three aspects:

- *Information-based interoperability*, which is usually discussed from a view point of communication or interaction standards, such as TCP/IP, XML and SOAP etc.,
- *Resource-based coordination*, which focuses on the controlling and scheduling of the sharing resources such as employees, machines and inventory etc.,
- *Business rules-based collaboration*, which focuses on the mechanisms of process coordination, such as partnership trust and conformance assessment.

As a basis for discussion, a conceptual model for business process collaboration is developed in Section 2 to illustrate a general motivating scenario and to find out the underlying problems, which lead to conclusion that both process description model and performance model should be developed to meet the requirement of process collaboration.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 191–204, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Since business process modeling plays a fundamental role in process management, a brief review of business process modeling methods will be given in Section 3. So far, there has been no universally accepted process modeling standard that can satisfy all collaboration requirements, now the research trend is to reconstruct current process modeling methods by using an ontology that standardizes a shared vocabulary for communication and makes the semantics in the collaboration explicit.

Those approaches to facilitate business process collaboration, such as workflow interoperability, web service choreography and ambient intelligence, all involve semantic interoperability as a key factor. In section 4, we focus on the concept of interoperability and discuss the related research issues.

Currently, the semantic web technologies have become the most promising direction for integration and collaboration. Based on the above review and analysis, we believe semantic web technologies could help to enhance the efficiency and effectivity of process collaboration. So a framework of process interoperability based on semantic agent is given in Section 5. Key technologies such as process ontology and agent coordination rules are also discussed.

In a word, the aim of this paper is two-fold. On the one hand, it is to give an overview of recent research efforts and future trends related to business process collaboration; on the other hand, it is to propose our idea of using the semantic agent to facilitate business process collaboration in both application centric and human centric process environment.

2 Business Process Collaboration

Over the past ten years, under the huge competition pressure on cost, quality, service and time, the value of process management has been recognized by most enterprises. Many ideas, such as BPR (Business Process Reengineering), CPI (Continuous Process Improvement) or TQM (Total Quality Management) are discussed to show that process management can play a crucial role in creating sustainable competitive advantage.

The rapid development of the Internet and web infrastructures in the last few years has brought fundamental changes and enormous opportunities in the way business patterns are made available to both individuals and organizations. With the trend toward increasing globalization of manufacturing and outsourcing of functions to external partners, the challenge for the next years will be moving from intra-organization process management to inter-organizational process interaction, coordination and collaboration in the global supply chains.

2.1 A Conceptual Model of Business Process Collaboration

To describe the general situation of business process collaboration among different stakeholders, a basic conceptual model was created as Fig. 1.

Sets A and B represent two different processes: $A = \{a_1, a_2, \cdots, a_n\}, B = \{b_1, b_2, \cdots, b_m\}$, where a_i or b_j is an activity. We use different modeling forms

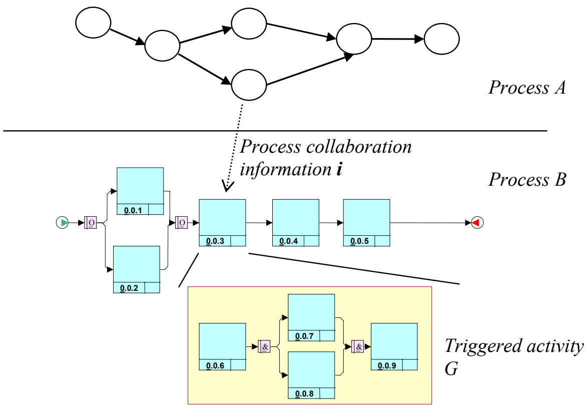


Fig. 1. Conceptual Model of Business Process Collaboration

to illustrate different processes because in industrial practice, various process modeling methods could be adopted by different modeling developers or users.

For the reason such as resource sharing or product assembling, process A and process B need to coordinate across manual or automated activities to achieve a common business goal. In this scenario, the running status of process B needs to be controlled or adjusted dynamically according to the running status of process A. Hence we must choose some key activities of process A, the information occurred by these activities would be treated as key performance indicators to control process B in a collaboration environment, which was called *Process Collaboration Information*, represented by set I, $I = \{i_1, i_2, \cdots, i_l\}$. By capturing and "translating" this information through specialized interface, process B could adjust its status of running automatically: For instance, trigger a set of activities G, to meet the collaboration requirement of both parties.

This conceptual model describes a most common situation of business process collaboration. It is specially pointed out that the concept of *Process Collaboration Information* is used to represent a general situation. In different collaboration environment (e.g. the degree of task automation, different process structures), collaboration information can be just one simple message about a single process entity, or can be composed by complex statistical performance indicates about the whole activity chain. Hence the main challenge for business process collaboration is *How to describe the process* and *How to construct Process Collaboration Information.*

2.2 Approaches to Business Process Collaboration

In the last decades, there has been a lot of research on business process modeling, software, architectures and standards to address business process integration in both academic and industrial areas. Thus interoperability among various process representation methods and heterogeneous process management systems has been an emerging need. From a modeling point of view, efforts can be carried out from two aspects as the conceptual model pointed out:

- *Process Description Model.* In essence, process description is a form of business knowledge that facilitates the understanding and communication of industrial users. There has been a lot of business process modeling techniques and tools (Section 3) which emphasize different perspectives for their own purposes. Hence in a collaboration environment, there must be a common business process description method and its schema language for distributed stakeholders to understand and communicate in a standard way[1].
- *Process Performance Model.* Given the complexity of business process, the process performance model is to represent the process performance indicators and their relationships, and more importantly, to provide the necessary information which helps collaboration and communication among different processes. In industrial practice, each process, especially in inter-organization environment, could be seen as separate profit and decision center, the conflict of single process goal and the whole collaboration goal should be solved on business level.

Furthermore, from a technical point of view, there must exist a standard description format and access protocols for both collaboration parties to publish their ability through a uniform interface, which is usually called workflow interoperability or web service choreography, so that process tasks can automatically be executed among potential partners with the help of functions provided by web service.

A number of initiatives is presently carried out worldwide by several academic and industrial research groups in some related areas to support business process collaboration for both human and automated tasks. One such trend is process integration by *Web Service* as a universal computing platform. In this area, process management is usually used together with another concept as *workflow* technology, of which the aim is "automation of a business process" [2]. The new research area merged into with workflow and web services merged, called workflow interoperability or web services choreography [3], aims at providing dynamic trading service (e.g., electronic purchase order) to business collaboration partners with universal description language and protocols.

On the other hand, not all process tasks are high structured and can be executed automatically by machine (e.g., human decision). Another research area is *Ambient Intelligence* (AmI), which is a human centered technology that is intuitive to the needs and requirements of the human actors. They are nonintrusive systems that are adaptive and responsive to the needs and wants of different individuals. AmI is a new paradigm in the area of information communication technology (ICT), context and context-awareness are central issues to AmI. There are many existing systems applying AmI/context aware technologies in the office, shopping store, house, hostel, museum, etc., but still very few systems exist at the moment to support the manufacturing collaboration environment. One of such systems is *iShopFloor* [4], which is an Internet-enabled agent-based intelligent system that provides an open architecture for distributed intelligent manufacturing process planning, scheduling, sensing, and control of the shop floor.

3 Business Process Modeling

Business process is defined [5] as “a structured, measured set of activities designed to produce a specified output for a particular customer or market”, and may be defined based on three dimensions as below: entities, objects and activities. Process models, which aim at a common understanding and analysis of business processes, play a crucial role in the implementation of any process improvement projects such as business process reengineering (BPR) or continuous process improvement (CPI). Literature review shows that process models should provide functions mainly covering two fields:

– *Description.* In essence, process representation is a form of business knowledge, to facilitate the understanding and communication of industrial users.

– *Analysis.* For the purpose of increasing process efficiency, not only representation is needed, models should also provide proper methods to analyze process and support process design/re-design, which are the core and most difficult task of BPR and CPI projects.

As process is usually defined as “a set of activities arranged in a specific order”, most current process modeling techniques describe processes in the form of activities and other process variables, such as entities, resources and objects, and their relationships, such as time series, logic and hypotaxis. These models, usually created by graphical modeling tools, including Petri Net[6], RAD[7], EPC[8], UML[9] and IDEF3[10] etc., have significant advantages on simplicity and descriptive ability. However, they fall short in analyzing capability to assist enterprise users with process designing and execution, mainly because of the following reasons:

– Graphical representation based on informal notation lacks mathematical accuracy and formal semantics, which makes it difficult to take effective analysis of process models, and difficult to communicate, share and reuse as well.

– Besides time series and logical relationship of activities, the underlying nonlinear cause-and-effect relationship between process variables is seldom taken into consideration, which is important for disclosing the interrelationship between these controllable variables and performance improvement of process.

To meet the requirement of process collaboration among different stakeholders, process modeling methods should provide a common meta model and its associated schema language. The Process Specification Language (PSL) [11] has made significant effort to solve the problem. The goal of PSL is to create a process representation that is common to all manufacturing applications, such as scheduling, process modeling, process planning, production planning, simulation, project management, work flow, and business process reengineering. Based on the PSL ontology, different stakeholders can describe their business processes by using shareable terminology.

Motivated by those early efforts to standardize process description by ontology such as PSL, researchers try to extend it with domain knowledge added to enable a more widely collaboration integration, for instance, the project M3PE[3] developed a process ontology (m3po)[12] to incorporate and unify the different existing workflow meta models and workflow reference models. This project is still under developing and its next step is ontology mappings and validations from different existing workflow systems.

4 Process Interoperability

4.1 Definitions of Interoperability

Interoperability is a concept addressed very early during the design, development and enhancement of distributed systems. However, it is difficult to find a precise and general definition of interoperability. There exist different definitions of interoperability from different points of view.

- "Generally, the word 'inter-operate' implies that one system performs an operation on behalf of another." [13]
- "The ability to communicate with peer systems and access the functionality of the peer systems." [14]
- From the software engineering point of view, interoperability means: two cooperating software systems can easily work together without a particular interfacing effort.
- "The ability of two or more software components to cooperate despite differences in language, interface, and execution platform." [15]
- "The ability to integrate data, functionality and processes with respect to their semantics." [16] And "interoperable" is identified as a high degree of compatibility.

4.2 Semantic Interoperability

[13] defines a simplified interoperability framework to describe the interaction between two enterprises, as shown in Fig. 2.

- *Business layer*: includes business environment, business processes;
- *Knowledge layer*: includes organizational roles, skills and competencies of employees, knowledge assets;
- *ICT layer*: includes applications, data and communication components.

[3] http://m3pe.org

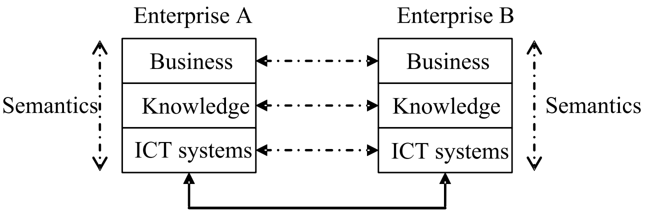


Fig. 2. Interoperability on all layers of an enterprise[13]

In the paper [17], a historical perspective and an overview of the interoperability is discussed as Fig. 3 shows. The *Remote Procedure Call* (RPC) and CORBA's *Interface Definition Language* (IDL) represent the early interoperability evolution. In the 90's, research has been focused on *signature level*, *protocol level* and *semantic level*. In an increasing order of complexity and difficulty, interoperability can be classified into four levels, which are physical, data, specification and semantic levels.

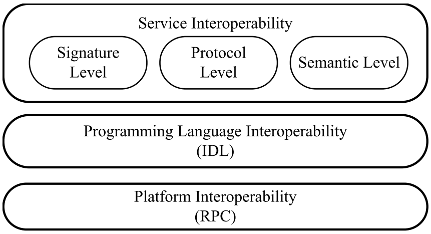

Service Interoperability
Signature Level
Protocol Level
Semantic Level
Programming Language Interoperability (IDL)
Platform Interoperability (RPC)

Fig. 3. Classic Levels of Interoperability[17]

According to [18], previous research in semantic interoperability can be categorized into three areas.

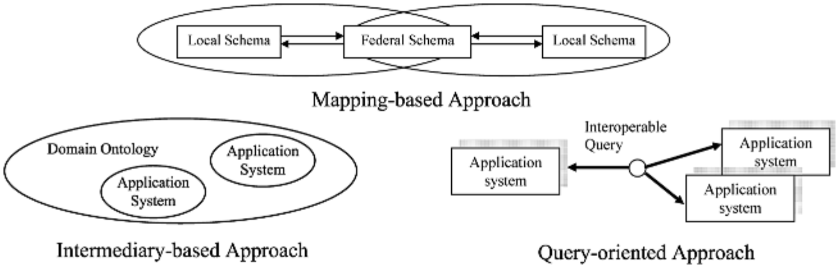


Fig. 4. Research Approach in Semantic Interoperability

– *Mapping-based*: construct mappings between different systems. The drawback of this method is that the mapping relationship is not designed to be independent of particular schemas and applications.
– *Intermediary-based*: use an ontology to share standardized vocabulary or protocols to communicate with each other. Its knowledge is domain-specific, but independent of particular schemas and applications.
– *Query-oriented*: based on interoperable languages, most of which are declarative logic-based language.

4.3 Workflow Interoperability Standards

Both standards for electronic data interchange and development of workflow systems have a longer history dating back to the 1970s , and process interoperability has been studied since the middle of the 1990. The latest research surge is emerging workflow management and web service into business integration scenario, which is called workflow choreography interface (or behavioural interface, abstract process, collaboration protocol profile, etc.)

A web service is a "software application identified by a URI, whose interfaces and bindings are capable of being defined, described and discovered by XML artifacts, and which supports direct interactions with other software applications using XML based messages via internet-based protocols."[19] Recent research in this area developed a lot of workflow interoperability and web services choreography standards, such as WSDL (Web Services Description Language)[20], SOAP (Simple Object Access Protocol)[21], UDDI (Universal Description, Discovery and Integration)[4] [22], WSFL (Web Services Flow Language)[23], BPEL4WS (Business Process Execution Language for Web Services). [24] and [25] have given a historical perspective of these standards.

5 A Process Collaboration Framework with Semantic Agent

With the emerging and rapid development of the semantic web, it is possible to adopt the novel semantic web technologies to help to enhance the efficiency and effectivity of process collaboration.

5.1 Semantic Web Technologies

The semantic web can be envisioned as an extension of the current web, which aims to make the web more understandable to computer programs, and allows data to be shared and reused across applications, enterprises, and community boundaries, easily.

There are two backbone technologies for the semantic web: RDF and OWL[26]. They, as web standards, provide a framework for asset management, enterprise integration, and sharing and reusing data on the web. These standard formats for

[4] http://www.uddi.org/

data sharing span different applications, enterprises, and community boundaries. All users - both human and machine - can share and understand the information available on the semantic web. The foundation of RDF[27] is built on a very simple model, but the basic logic can support large-scale information management and processing in a variety of different contexts. The assertions in different RDF files can be combined, providing far more information together than what they contain separately. RDF supports flexible and powerful query structures, and developers have created a wide variety of tools for working with RDF. OWL[28] provides a language for defining structured, web based ontologies. This delivers richer data integration and interoperability among descriptive communities. Many semantic web based information systems have been created[5], and have been successfully used in some industrial applications.

5.2 Application Scenario

The framework is developed to meet the requirement of project AMI-4-SME[6] in two main application scenarios: machine maintenance and shop floor control.

Machine Maintenance. Maintenance management is all about managing asset. It is defined as "the coordination, control, planning execution and monitoring of the right equipment maintenance activities of manufacturing operations"[29]. Maintenance is looked at from two perspectives, the first is providing the maintenance personnel with accurate and relevant realtime information on their machines in order to enable them to introduce and implement the appropriate strategies. The second perspective helps the staff carry out all necessary maintenance, both scheduled and unscheduled, in order to get the machines up and running in as quick a time as possible.

Combining the information coming in from the product tags with the machine sensors provides a vast amount of invaluable data. Detailed figures on machine utilization, mean time between failures, mean time between maintenance, equipment downtime, maintenance staff efficiency, overall equipment efficiency, and comparison of various maintenance strategies, to name but a few, can all be inferred from the collected realtime data. On the other hand, the maintenance technician, with the help of PDA, can get the information on the current state of the machine such as temperature, action that caused the failure, operator name etc. Details of the previous maintenance are also supplied to the maintenance technician.

Shop Floor. Shop-floor control is concerned with the efficient management and usage of resources at the lowest level of control on the shop floor of a manufacturing plant. The realtime information coming from the product manufacturing process provides exact locations for each of the batches of product. Should a certain machine fail then the batch location information can used to re-route

[5] Semantic Web Challenge. http://challenge.semanticweb.org

[6] http://www.ami4sme.org/

the effected batches as efficiently as possible in order to keep the impact of the machine failure to a minimum.

Information from various sources is used to control the processing of orders on the factory floor. It primarily relies on batch location information and also information from the employee roster, order book, stock room and others it controls how orders are routed through the factory and it also controls what workstations employees are working at. The data acquired from the tags and sensors can also be processed to provide the user with information on traceability, accountability and reliability. Traceability and accountability are of particular significance as they are necessary for meeting the guidelines laid down by standardization bodies i.e. ISO. Efficiency is useful in ensuring that the company is getting the most output of their assets.

5.3 Process Collaboration Framework Structure

Fig. 5 shows a process collaboration framework. Our idea is to transfer process collaboration information between different processes by semantic agents. All agents can access various business process management systems and capture information of different processes represented by process ontology.

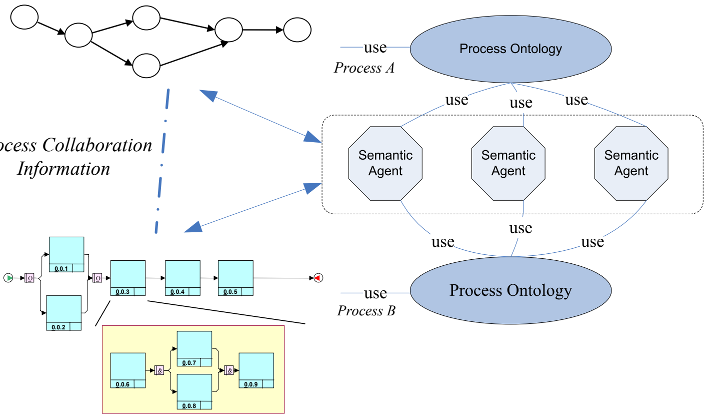


Fig. 5. Process Collaboration Framework Based on Semantic Agent

Ontology. Ontology is a general conceptualization of a specific domain in a both human and machine readable format. In general, it consists of classes, properties, relationships, and axioms. To realize interoperability and collaboration of different processes, it is necessary to build a formally and explicitly expressed process ontology, which can be classified into two categories:

- *Process Description Ontology* is to give formal semantics to traditional process modeling elements, such as entities, objects and activities, their relationships etc. For example, we can get details of some product by the statement:

```
<rdfs:Class rdf:ID="GetDesiredProductDetails">
<rdfs:subClassOf rdf:resource
        ="http://www.product.org/Process#AtomicProcess" />
</rdfs:Class>
```

PSL provides feasibility of extending its core ontology to represent most current process models with similarity-based ontology mapping. With Process Description Ontology, industrial users could get a common and formal understanding from their existing traditional process models and computers would be able to work cooperatively and efficiently in a collaboration environment.

– *Process Performance Ontology* is to express process performance for special collaboration partners with process domain knowledge added. For instance, in the maintenance scenario, the performance indicators such as mean time between failures, mean time between maintenance, equipment downtime, maintenance staff efficiency, and overall equipment efficiency will be given precise definition.

Semantic Interface. In this AmI scenario, process information could be collected by using three methods: *Tags, Machines Sensors,* and *PDAs* (Personal Digital Assistants).By applying tags (RFID, barcode, etc.) to the product component, the status of each production activity can be monitored in realtime throughout the whole production cycle. The type of tag used will be dependent on many factors such as the type of product, company size, complexity of production, production techniques etc. The production machinery is equipped with networked sensors which monitor the status of the machines (idle, off, processing product, failure). This information is available in realtime to the maintenance staff. Either maintenance or shop-floor staff having wireless PDAs equipped with tag readers could get information on particular product or machine to support their decision making. All the set of information sources and services needs to run the transaction protocol; for example, ontology searching, software-capability profiles, programming language, and the interoperation acts that will set up the semantic interfaces.

Agents and Rules. An agent is a software entity with intelligent properties, according to [30], the integration of agent technology and ontology will significantly affect the use of web services and the ability to extend programs to perform tasks for users more efficiently and with less human intervention. Agents act at the interface for the human-human and human-machine collaboration in business process integration. In this AmI scenario, different intelligent software agents (e.g., context agent, maintenance agent, production agent) work together to access heterogeneous information systems in anticipating user's requirements and thus avoid manual browsing for common information gathering tasks.

The shared process ontology allows for solving problems concerning heterogeneity of knowledge representation between distributed agents. On the other hand, intelligent agents that can automatically find any information requested by the user and can execute some intelligent issues like coordination, negotiation and agreement thus avoid inefficient or manual 'surfing'. With semantic agents any stakeholder will have instant access to all of distributed processes running at anywhere within the organization, regardless of format, structure, or location of the information.

With the help of *Process Performance Ontology*, it is possible to build a set of business rules (e.g., time, constraint, exception) to facilitate agent acting, reasoning, and coordinating with each other. The explicit description of process performance related information makes the process coordination feasible (e.g., resource conflict solving among different parts of the extended enterprise). This automated process reduces human intervention in process management, and thus enables them to focus on the most complex perspective. Ontology inference or reasoning, which can improve the efficiency of query and processing of innovation related instance data, will play the role in realizing the alignment analysis among different objects and rules, which could adopt Description Logic, FLogic or Horn Logic as theoretic foundation.

System Infrastructure. The key components of the system infrastructure include ontology server, rule base and web server as shown in Fig. 6. Through this infrastructure, Process Agents communicate with the Web Server to act and reference using the Process Ontology Database and the Process Rule Base, to realize collaboration between different processes.

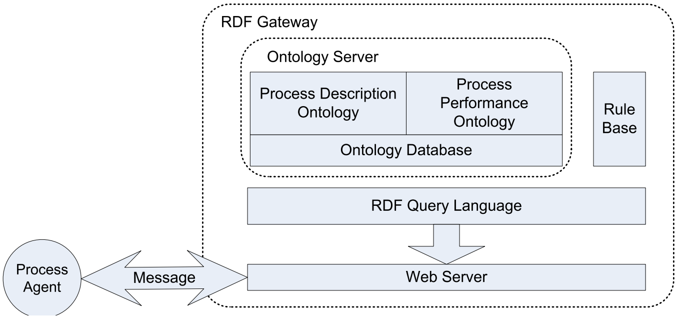


Fig. 6. System Infrastructure

RDF Gateway[31] is an ideal development environment for this infrastructure. It provides (1) RDF database to store Ontology in RDF triples, (2) the RDFQL language to query RDF and to execute server-side tasks, (3) certain inference capability (by RULEBASE command) to support RDFS and other customized rules, (4) the stand alone Web Server to communicate with Process Agents.

6 Conclusion

Business process collaboration plays a more and more important role in today's global manufacturing environment. Based on the conceptual model provided, we have discussed the requirements of business process collaboration, and reviewed some related research area, such as business process modeling and interoperability. These works have built a firm foundation for further research in this area.

Since the semantic web technologies have become the most promising direction for integration and collaboration, a business process collaboration framework based on semantic agents is provided. This framework needs to be further developed in detail. The further research issues include ontology mapping, agent design and coordination rules. The implementation work based on RDF Gateway is still going on. Our belief is that semantic web techniques could help to enhance the efficiency and effectivity of process collaboration.

Acknowledgements

This paper is based upon works funded by the European Union 6th Framework Specific Targeted Research Project (STREP) Ambient Intelligence for Systemic Innovation in Small to Medium Enterprises (AMI4SME). We would like to thank David Mulligan, Simrn Kaur Gill, Gearoid Hynes, Thai Vinhtuan and Fergal Monaghan for their key contribution of this research project. We also like to thank to Armin Haller and Eyal Oren for their introduction on project M3PE.

References

1. Dayal, U., Hsu, M., Ladin, R.: Business process coordination: State of the art, trends, and open issues. In The 27th VLDB Conference, Roma, Italy (2001)
2. Fischer, L.: The Workflow Handbook 2004. Future Strategies Inc. (2004)
3. Zhao, J.L., Cheng, H.K.: Web services and process management: a union of convenience or a new area of research? Decision Support Systems, 40(1) (2005) 1–8
4. Shen, W., Lang, S., Wang, L.: ishopfloor: an internet-enabled agent-based intelligent shop floor. IEEE Transactions on Systems, Man and Cybernetics, Part C, 35(3) (2005) 371–381 TY - JOUR.
5. Davenport, T., Short, J.: The new industrial engineering: Information technology and business process redesign. Sloan Management Review, 1990 Summer (1990) 11–27
6. Pinzon, L.E.: Petri Net Models. http://rutcor.ruters.edu/ pinzon/papers/rrr1/node15.html (2006)
7. Ould, M.A.: Business Process - Modelling and Analysis for Reengineering and Improvement. Wiley, New York (1995)
8. Scheer, A.W.: ARIS - Business Process Frameworks. Springer, Berlin (1999)
9. Fowler, M., Scott, K.: UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley (2000)
10. KBSI: IDEF3 Process Flow and Object State Description Capture Method Overview - Process Description Capture Method.
11. Gruninger, M., Menzel, C.: The Process Specification Language (PSL) theory and applications. AI Magazine, 24(3) (2003) 63–74
12. Haller, A., Oren, E.: A process ontology to represent semantics of different process and choreography meta-models. Digital Enterprise Research Institute(DERI), Galway, Ireland (2006)

13. Chen, D., Doumeingts, G.: European initiatives to develop interoperability of enterprise applications-basic concepts, framework and roadmap. Annual Reviews in Control, 27(2) (2003) 153–162
14. Vernadat, F.: Enterprise modeling and integration: principles and application. Chapman & Hall, London (1996)
15. Wegner, P.: Interoperability. ACM computing surveys, 28(1) (1996) 285–287
16. IEC TC 65/290/TC (2002), Industrial process measurement and control
17. Strang, T., Linnhoff-Popien, C.: Service interoperability on context level in ubiquitous computing environments. In International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet (SSGRR2003w), L'Aquila, Italy (2003)
18. Park, J., Ram, S.: Information systems interoperability: What lies beneath? ACM Transactions on Information Systems, 22(4) (2004) 595–632
19. W3C: Web services architecture requirements. (2003)
20. W3C: Web services description language (WSDL) 1.1. (2001)
21. W3C: Simple object access protocol (SOAP) 1.1. (2000)
22. Zimmermann, O., Tomlinson, M.R., Peuser, S.: Perspectives on Web Services : Applying SOAP, WSDL and UDDI to Real-World Projects. Springer (2003)
23. Leymann, F.: Web services flow language (WSFL 1.0). Technical report, IBM (2001)
24. Andrews, T., et al.: Business process execution language for web services version 1.1. IBM, BEA Systems, Microsoft, SAP AG, Siebel Systems (2003)
25. zur Muehlen, M., Nickerson, J.V., Swenson, K.D.: Developing web services choreography standards–the case of rest vs. soap. Decision Support Systems, 40(1) (2005) 9–29
26. Miller, E.: The WBC's Semantic Web Activity: an update. Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications], 19(3) (2004) 95–97
27. RDF Primer. http://www.w3.org/TR/rdf-primer (2006)
28. OWL. http://www.w3.org/2004/OWL (2006)
29. de Vries, J.: Maintenance management. http://www.managementsupport.com (2000)
30. Hendler, J.: Agents and the semantic web. Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications], 16(2) (2001) 30–37
31. RDF Gateway. http://www.intellidimension.com (2006)

An Ontology Architecture for Integration of Ontologies

Jeongsoo Lee[1], Heekwon Chae[1], Kwangsoo Kim[1,*], and Cheol-Han Kim[2]

[1] Dept. of Industrial & Management Engineering, Pohang University of Science and Technology, San 31, Hyojadong, Namgu, Pohang, South Korea, 790-784
{jsrhyme, hkchae, kskim*}@postech.ac.kr
[2] Dept. of Information System Engineering, Daejeon University, 96-3, Yongwoondong, Donggu, Daejeon, South Korea, 300-716
chkim@dju.ac.kr

Abstract. Ontologies are expected in various areas as promising tools to improve communication among people and to achieve interoperability among systems. For communications between different business domains, building an ontology through integrating existing ontologies is more efficient way than building the ontology without them. However, integration of ontologies is very struggling work since languages, domains, and structures of ontologies are different from each other. In this paper, we suggest an Ontology Architecture which solves this problem by providing a systematic framework to classify ontologies from three kinds of viewpoints: language, domain range, constructs. The Ontology Architecture consists of three axes according to the three viewpoints: Ontology Meta Layering axis, Semantic Domain Layering axis, and Ontology Constructs Layering axis. Because three axes in the Ontology Architecture are designed to improve the syntactic and semantic interoperability among ontologies, the integration of ontologies can be readily achieved.

1 Introduction

Ontologies are used to improve communication either among humans or among computers by specifying the semantics of the symbolic apparatus used in the communication process. Using ontologies to improve communication means that there is a common ontology and that all stakeholders share the ontology. Actually, many researchers and developers endeavor to build their own standardized ontologies in various areas. However, while one can expect utilizing such existing standard ontologies for communication and interoperability in a specific domain, building a new shared ontology must be required for collaboration among different domains. And building a new ontology can be done in an even more effective way through integrating existing ontologies[12].

Integrating ontologies includes following steps[17]: identifying candidate ontologies, analyzing those ontologies, and applying integration operations. However, because ontologies can be made regarding all the areas and things in the real-world as target domains, ontologies have various characteristics and forms by nature. Therefore, to accomplish above mentioned steps for ontology integration, interoperability among ontologies which have such various characteristics and forms should be

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 205 – 211, 2006.
© Springer-Verlag Berlin Heidelberg 2006

achieved. That is, guaranteeing interoperability among ontologies for integration is a key factor for building ontologies efficiently. To achieve the interoperability among ontologies, layering criteria to evaluate and analyze existing ontologies are required. In fact, a number of studies are done to suggest layering methodologies from their own viewpoints. However, most studies suggest "partial" layering criteria reflecting only one aspect of ontologies rather than considering various aspects of ontologies on the whole. Applying such partial layering criteria cannot lead to complete interoperability among ontologies because several important aspects may be overlooked. Incomplete interoperability may incur not-intended result such as syntactic or semantic incompleteness or structural heterogeneity when integrating existing ontologies to build a new shared ontology.

In this paper, we suggest an ontology architecture which harmonizes systematically representative three aspects of ontologies that should be considered for interoperability among ontologies. The ontology architecture facilitates integration of ontologies through making integration process flexible and accurate.

This paper is organized as follows. In section 2 we introduce three layering criteria which make above three axes in the suggested ontology architecture respectively. Section 3 provides and illustrates the ontology architecture which integrates the three axes. In section 4, we introduce related works, and finally section 5 concludes our works.

2 The Three Layering Criteria

In this section, the three layering criteria for each axis and detailed layers for each axis of the suggested ontology architecture are introduced.

2.1 Ontology Meta-layering based on MDA

Model-Driven Architecture(MDA)[13] is a development framework proposed by Object Management Group(OMG). MDA supports the development of component- and service-based software systems through modeling techniques based on notations such as Unified Modeling Language(UML)[16]. MDA is based on meta-modeling which classifies models and modeling languages into several layers systematically to facilitate the development through modeling. MDA adopts the four-layer meta-modeling architecture that consists of meta-meta-model(M3) layer, meta-model(M2) layer, model(M1) layer and instance(M0) layer. Refer to MDA specifications for detailed explanations about the four-layer architecture.

MDA and its four-layer architecture provide a solid basis for defining meta-models of any modeling language[5]. That means that interoperability among different modeling languages can be achieved solidly through the four-layer architecture. So it is the straight choice to apply the four-layer architecture for ontology modeling languages for interoperability. In fact, there were several studies for applying the four-layer architecture of MDA to the ontology area[5, 6, 8], and OMG also tries to define an ontology meta-model based on MOF and the four-layer architecture[14].

This paper also adopts the four-layer meta-modeling architecture as following Fig. 1, and introduces this meta-layering as an axis of the ontology architecture.

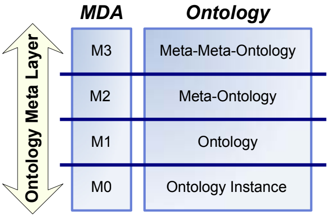


Fig. 1. Meta-layering of ontologies

2.2 Semantic Domain Layering

Ontology layering by semantic domain ranges is derived from the needs of an upper ontology for integration of ontologies. Every ontology has a target domain and the target domain has restricted range. To enhance semantic sharing across different ontology domains, semantic interoperability across the ontologies should be supported. One of the approaches to supporting semantic interoperability among ontologies is to use upper ontologies[10]. Upper ontologies provide definitions for general-purpose terms only[11], while its domain range may be as wide as whole real-world. And the main purpose of upper ontologies is to act as a foundation for more specific domain ontologies[11]. There are several researches about upper ontologies are now underway: Suggested Upper Merged Ontology(SUMO)[18], Upper Cyc Ontology[15], a Descriptive Ontology for Linguistic Cognitive Engineering[1].

In our study, we extended concepts of upper ontologies to introduce semantic domain layering as another axis of the ontology architecture. The semantic domain axis is layered as following Fig. 2.

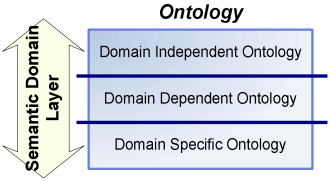


Fig. 2. Semantic domain layering of ontologies

The idea of domain independent ontologies is same as that of upper ontologies. A domain independent ontology provides basic concepts and relations which are adopted to build domain dependent ontologies and domain specific ontologies. Domain independent ontologies are intended to be fundamental and universal to ensure generality and expressivity for a wide range of domains.

As same as mid-level ontologies defined in SUO, a domain dependent ontology serves as a bridge between a domain independent ontology and a domain specific ontology. Domain dependent ontologies provide basic concepts and relation to build domain specific ontologies as domain independent ontologies do. However domain dependent ontologies provide more concrete and special concepts than domain independent ontologies.

A domain specific ontology specifies concepts particular to a domain of interest and represents those concepts and their relations from a domain specific perspective. Domain specific ontologies may be built by importing or extending domain independent and depedent ontologies. Semantic sharing by reusing common domain independent and dependent ontologies improves semantic interoperability between different ontologies.

2.3 Ontology Construct Layering

There is an agreement in researches about ontologies that basic constructs of ontologies include concepts and their relations. Logics are added to express more clear relationships between the concepts and relations[3]. Logic-based ontologies are ontologies in which the logics are included, and Non-logic-based ontologies don't include the logics. It depends on the objective of an ontology whether logics are included in the ontology. That is, additional constructs can be added to the basic constructs of ontologies according to objectives of ontologies. Because different ontologies may have different additional constructs, classification of constructs of ontologies is needed for interoperability among the ontologies. In this paper, we classified constructs of ontologies as five layers as illustrated in Fig. 3 to introduce ontology construct layering as the other axis of the ontology architecture.

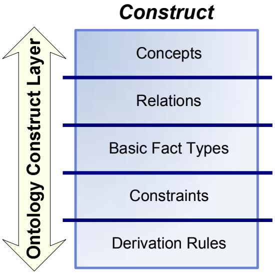


Fig. 3. Construct layering of ontologies

The constructs of ontologies are explained as follows.

- Concepts: A concept defines a basic and abstract idea that is commonly used in an ontology domain. It is represented as a word or phrase.
- Relations: A relation describes a way in which two or more concepts are interrelated; usually described by a verb or verb phrase.
- Basic fact types: A basic fact type is a kind of primitive sentence or fact. It is composed of concepts and relations. If it is always true in the ontology containing it, it can play a role as an axiom in the logic-based ontology.
- Constraints: A constraint is the restriction that is applied to a fact type.
- Derivation rules: rules, functions or operators (including mathematical calculation or logical inference) to derive new facts from existing facts.

3 The Ontology Architecture

Each of three layering approaches explained in section 2 provides several advantages to integration of ontologies individually.

Ontology meta-layering, which is for syntactic aspect of ontologies, acts as an implementation-level basis for ontology integration by providing interoperability among different ontology representation languages. For example, if two ontologies are built using Web Ontology Language(OWL), one of the ontology representation languages, and Topic Maps, another one of them, respectively, we can expect guaranteed interoperability between the two languages by using various methodologies about transforming between the two languages through M2 layer[4, 7, 20]. Semantic domain layering reduces complexity of ontology integration and building process by improving reusability of ontologies. Ontology construct layering derives integration classified by constructs. This makes ontology integration process more intuitive work. And by considering ontology construct layering, constructs which are not required for integrated ontology to be built can be easily selected and excluded. This improves efficiency of ontology integration.

In this paper, we suggest an ontology architecture as a basic guide for considering all the three layering approaches comprehensively. As following Fig. 4, the suggested ontology architecture harmonizes semantic domain layering and ontology constructs layering for semantic interoperability on the syntactic basis of ontology meta-layering.

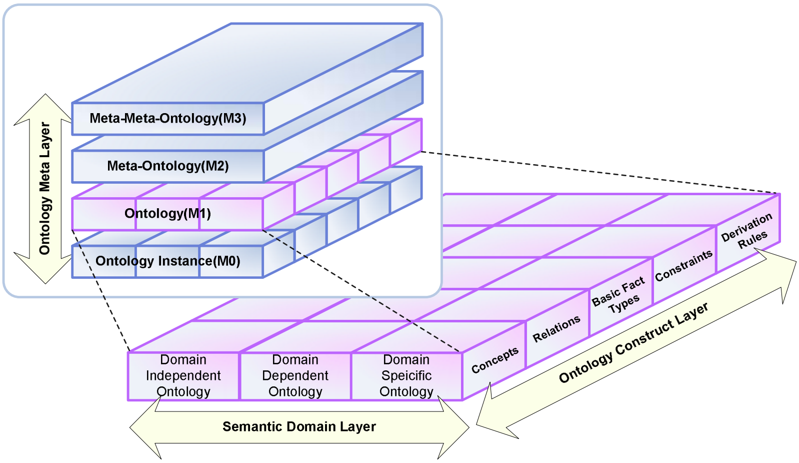


Fig. 4. The Suggested Ontology Architecture

The ontology architecture is expected to act as a map for identifying ontologies or a guideline for integration of ontologies. That is, the characteristics of an existing ontology or a new ontology can be grasped at a glance using the ontology architecture. This makes the ontology architecture to play important roles in the essential processes for ontology integration such as identifying candidate ontologies and analyzing those ontologies.

4 Related Works

The suggested ontology architecture is to support integration of ontologies by providing interoperability among ontologies. Interoperability among ontologies can be classified as two categories: syntactic interoperability and semantic interoperability. Related works also can be classified as the two categories.

Most studies related to syntactic interoperability among ontologies originate from MDA and its four-layer meta-modeling as Bezivin et al.[2], Duric et al.[5], Herre et al.[8] did. Bezivin et al. proposed a methodology for interoperability between modeling technical space and ontology technical space through M3 layer after classifying ontologies as four layers like MDA. Duric et al. identified about the location of ODM in meta-layer and discussed about interoperability between ODM and OWL. Herre et al. suggested three-layered meta-ontology architecture introducing Abstract Core Ontology(ACO).

The studies about the semantic interoperability include the methodologies about layering of domain ontologies by adopting the idea of upper ontologies. SUMO[18], Upper Cyc Ontology[15], DOLCE[1] are those studies. MITRE[10] surveyed and evaluated several upper ontologies to discuss about their application to military domain of U.S. government. Additionally Kent[9] suggested a methodology about building a global ontology for temporal integration of ontologies.

As a study which considered both of two aspects of interoperability, there is Standard Upper Ontology Information Flow Framework(SUO-IFF)[19]. But meta-layer of SUO-IFF pursues logical interoperability rather than syntactic interoperability as stated in the study of Herre et al.[8].

The above stated works did not support interoperability among ontologies comprehensively, since their layering criteria focus on restricted aspects of ontologies only. As a result, incomplete integration of ontologies may be caused.

5 Conclusions

Building an ontology through integrating existing ontologies is more efficient way than building the ontology without them. For the integration of ontologies, interoperability among ontologies is essential. To support the interoperability among ontologies, we suggested an ontology architecture which harmonizes representative three aspects of ontologies systematically. The suggested ontology architecture consists of three axes according to the three aspects of ontologies: ontology meta-layering axis, semantic domain layering axis, and ontology constructs layering axis.

The ontology architecture provides: (i) syntactic interoperability among heterogeneous ontology representation languages through syntactic layering which corresponds to meta-layering axis, (ii) semantic interoperability and simplicity and efficiency of ontology integration process through semantic layering which corresponds to semantic domain layering and ontology construct layering, and (iii) contribution to ontology integration by acting as a map for identifying ontologies. From these advantages, the ontology architecture is expected to provide a solid basis for studies about ontology integration.

Acknowledgement

This work is supported by grant No. R01-2003-000-10171-0 from the Basic Research Program of the Korea Science and Engineering Foundation.

References

1. A Descriptive Ontology for Linguistic and Cognitive Engineering Website, http://www.loa-cnr.it/DOLCE.html.
2. Bezivin, J., Devedzic, V., Djuric, D., Favreau J., Gasevic, D., Jouault, F., “An M3-Neutral infrastructure for bridging model engineering and ontology engineering”, In Proceedings of the Interoperability of Enterprise Software and Applications(INTEROP-ESA’05), Geneva, Switzerland, February 2005, pp 159-172.
3. Bittner, T., Donnelly, M., Winter, S., “Ontology and Semantic Interoperability”, In book: Large-Scale 3D Data Integration, CRCPress, London, 2005.
4. Cregan, A., “Building Topic Maps in OWL-DL”, In Proceedings of the Extreme Markup Languages 2005, Montreal, Canada, August 2005.
5. Duric, D., Gasevic, D., Devedzic, V., “Ontology Modeling and MDA”, Journal of Object Technology, Vol. 4, No. 1, January-February 2005, pp 109-128.
6. Duric, D., “MDA-based Ontology Infrastructure”, Computer Science and Information Systems, Vol. 1, No. 1, February 2004, pp 91-116.
7. Garshol, L., “Living with topic maps and RDF”, Ontopia Whitepaper, http://www.ontopia.net/topicmaps/materials/tmrdf.html#N69, March 2003.
8. Herre, H., Loebe, F., “A Meta-ontological Architecture for Foundational Ontologies”, Lecture Notes in Computer Science, Vol. 3761, 2005, pp 1398-1415.
9. Kent, R., “The Information Flow Foundation for Conceptual Knowledge Organization”, In Proceedings of the 6th International Conference of the International Society for Knowledge Organization(ISKO), Toronto, Canada, August 2000, pp 111-117.
10. MITRE, “Toward the Use of an Upper Ontology for U.S. Government and U.S. Military Domains: An Evaluation”, MITRE Technical Paper, http://www.mitre.org/work/tech_papers/tech_papers_04/04_0603/04_1175.pdf, September 2004.
11. Niles, I., Pease, A., “Towards a Standard Upper Ontology”, In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems(FOIS-2001), Ogunquit, USA, October 2001, pp 2-9.
12. Noy, N., Hafner, C., “The State of the Art in Ontology Design – A Survey and Comparative Review”, AI Magazine, Vol. 36, No. 3, 1997, pp 53-74.
13. Object Management Group, “MDA Guide Version 1.0.1”, OMG Document: omg/2003-06-01, http://www.omg.org/docs/omg/03-06-01.pdf, June 2003.
14. Object Management Group, “Ontology Definition Metamodel Request For Proposal”, OMG Document: ad/2003-03-40, http://www.omg.org/docs/ad/03-03-40.pdf, August 2005.
15. OpenCyc Website, http://www.opencyc.org/.
16. Pahl, C., “Layered Ontological Modelling for Web Service-Oriented Model-Driven Architecture”, Lecture Notes in Computer Science, Vol. 3748, 2005, pp 88-102.
17. Pinto, H., Martins, J., “A Methodology for Ontology Integration”, In Proceedings of the 1st international conference on Knowledge capture, Victoria, Canada, 2001, pp 131-138.
18. Suggested Upper Merged Ontology Website, http://ontology.teknowledge.com/.
19. The SUO Information Flow Framework Website, http://suo.ieee.org/IFF/.
20. Vatant, B., “Ontology-driven Topic maps”, In Proceedings of the XML Europe 2004, Amsterdam, Netherlands, April 2004.

Automatic Alignment of Ontology Eliminating the Probable Misalignments

Seddiqui Md. Hanif, Yohei Seki, and Masaki Aono

Knowledge Data Engineering laboratory,
Department of Information and Computer Sciences,
Toyohashi University of Technology
hanif@kde.ics.tut.ac.jp, {seki, aono}@ics.tut.ac.jp

Abstract. This paper describes a novel approach of detecting misalignment at the time of aligning two different ontologies, and of eliminating the misalignments. Our objective is to reduce limitation of a specific technique of ontology alignment. Two aligned sets extracted by different alignment techniques from the same pair of ontology, are fed to the misalignment detection and elimination process to produce better alignments. Our experiments demonstrate that our method, taking advantage of misalignment detection and elimination, shows a good recall and precision.

Keywords: Ontologies, Semantic Integration and Interoperability, Ontology alignment.

1 Introduction

Semantic web technology is an extension of current World Wide Web consisting of machine understandable metadata, better enabling machine and people to work in co-operation. Ontology is an important part of semantic web which gives the metadata meaning by containing concepts, properties and relationships among concepts and properties. However, defining a full-fledged ontology is not only a difficult task, but also impractical in general web. Small domain ontologies rather easy to be created and managed even by a single user. But a problem arises when a user wants to communicate different web sites which are annotated by different ontologies created by different persons with different social factors like culture, thoughts and philosophy. Thus, to communicate different web sites consisting different ontologies, users must need to align the entities like concepts, properties etc. of ontologies. Finding adequate relationships between entities belonging to different ontologies is called "ontology alignment" [4].

Although ontologies are well-defined with a sophisticated language like OWL[1], but still there may be same words with different meaning due to polysemy, different words with same meaning as synonym, to some extent similar meaning with different words like hyponymy or hypernymy relations between words and above all there is some localized spelling and meaning of words. The lexical database WordNet[2] contains around 210,000 concepts. In the average each word has greater than 2.5

[1] http://www.w3.org/2004/OWL/

[2] http://www.cogsci.princeton.edu/~wn/

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 212 – 218, 2006.
© Springer-Verlag Berlin Heidelberg 2006

polysemous meaning, i.e same word but meaning is different [6, 9]. Thus, it may be easy to comprehend that same word with different meaning is frequently used. So, we must do something more than just adding alignment retrieved from lexical analysis and alignment retrieved from structural similarity. Again, the target user domain of semantic web is autonomous intelligent programs like intelligent agents in addition to human. So, probable misalignment detection and elimination methodologies will play an important role in automatic ontology alignment.

There are many ontology alignment tools available now-a-days like ‘Falcon’ [7, 8], ‘FOAM’, ‘CMS’, ‘OLA’, ‘CtxMatch 2’ [3] and so on. They all measure the similarity values for a pair of entities of different ontologies using string manipulation or structure evaluation to produce alignment.

Primarily, the misalignment detection and elimination is implemented as a part of an ontology alignment system proposed as OntoKDE[3]. Our system, OntoKDE also extracts terminological similarity by string manipulation and structural similarity using graph and relation available in the ontologies. Moreover, it is capable to detect misalignments and eliminate the misalignments already detected.

The formation of this paper is descried below. Section 2 focuses on the used methods of computing similarity which is the primary and principle steps of ontology alignment. Section 3 is used for describing methodologies to detect and to eliminate misalignments. Section 4 includes the concluded remarks.

2 Methodologies of Computing Similarity

Similarity computation methodologies [2] were divided into two categories in our system, terminological similarity computation and structural similarity computation [5]. The techniques of both categories computed similarity values between pair of entities of different ontologies. Similarities between homogeneous entities were computed, i.e. each class was compared against classes and each property was compared against properties.

Terminological similarity measures consider the isolated entities of ontology while structural similarity measures target the structure.

2.1 Terminological Similarity Measure (TSM)

Terminological similarity measures were based on linguistic methods of string-based and lexicon-based analysis. Entities of ontology contained URI and annotation properties like labels, comments etc. This information was analyzed for measuring terminological similarity in ontology level.

For terminological similarity measures, component words of an entity were decomposed. Stop words were eliminated and then stem form was extracted from inflected word form using slightly modified Porter stemming algorithm [13].

Similarity was measured between a component of one entity and a component of another entity by using direct string comparison, if it was not equal, then lexicon based similarity using synonym-set of WordNet was applied, if any of the two components was not present in WordNet, lexical similarity using Levenshtein’s edit

[3] OntoKDE is implemented by the authors of this paper at Knowledge Data Engineering laboratory in Toyohashi University of Technology, Japan.

distance [10] was measured for component strings as stated in [11]. The similarity scores between best matched components were then aggregated to get the similarity between entities. WordNet was also applied to measure the relatedness of entities using hypernymy, hyponymy and polysemy relationships.

2.2 Structural Similarity Measure (SSM)

The entities of ontology were organized in a graph according to the hierarchy defined in the ontology file. Nodes of the graph were entities. They were compared between two ontology-graphs considering direct children, all children and leaf-set.

Considering the postulate that two non-leaf elements are structurally similar if their children set are highly similar, or their immediate children sets are highly similar, or their leaf sets are highly similar [1]. The best score of one-to-one similarity values between children of different non-leaf elements of different ontology are propagated towards the parent-pairs [12].

The similarity between two non-leaf elements from different ontologies is calculated as:

$$Sim\ (p,\ q) = |\,aligned\ (c_{p,}\ c_q)| \,/\, (|c_{pa}|+|c_{qa}|-|\,aligned\ (c_{p,}\ c_q)|), \quad (1)$$

where p and q are non-leaf entities having c_p and c_q children-sets of p and q respectively, $|c_{pa}|$ is the number of children of p which is aligned, $|c_{qa}|$ is the number of children of q which is aligned and $|aligned(c_{p,}\ c_q)|$ is denoted by the number of alignment between children of p and children of q.

The same process was applied for measuring the similarity of non-leaf elements of two ontologies considering leaf elements.

3 Misalignments

An alignment is defined as a = (e_{o1}, e_{o2}, aligned-relationship | $e_{o1} \in$ ontology-1 and $e_{o2} \in$ ontology-2 $\wedge$ aligned-relationship is one-of (=, $\sqsubset$, $\sqsupset$)), and alignment set, A is a set consisting of alignments a_1, a_2....a_n. For a single ontological alignment system, in OntoKDE, terminological alignment set, A_T and structural alignment set, A_S are extracted separately. Then misalignment set, A_m is defined as

$$A_m = (A_S \cup A_T) - (A_S \cap A_T). \quad (2)$$

That is, misalignment is the set of the alignments which are not commonly present in both alignment set, A_S and $A_{T.}$

The overall system of misalignment detection and elimination is shown in the Figure 1. The module can be applied not only within ontology alignment module, but also inter-ontology alignment module.

Two sets of alignment, extracted by different methods within same ontology alignment module or received from completely different ontology alignment systems, are fed to misalignment detection and elimination module. Within the module, two sets are compared and detected misalignment. The common alignment A_c, which is not considered as misalignment, is considered as a part of final alignment-set. The detected misalignment set is fed to misalignment elimination sub-module. Through the recalculation of similarity and filtering process a misalignment is judged and detected most relevant alignments, A_{mf} from misalignment set A_m. A_c and A_{mf} are summed for resulting final alignment [Figure 1].

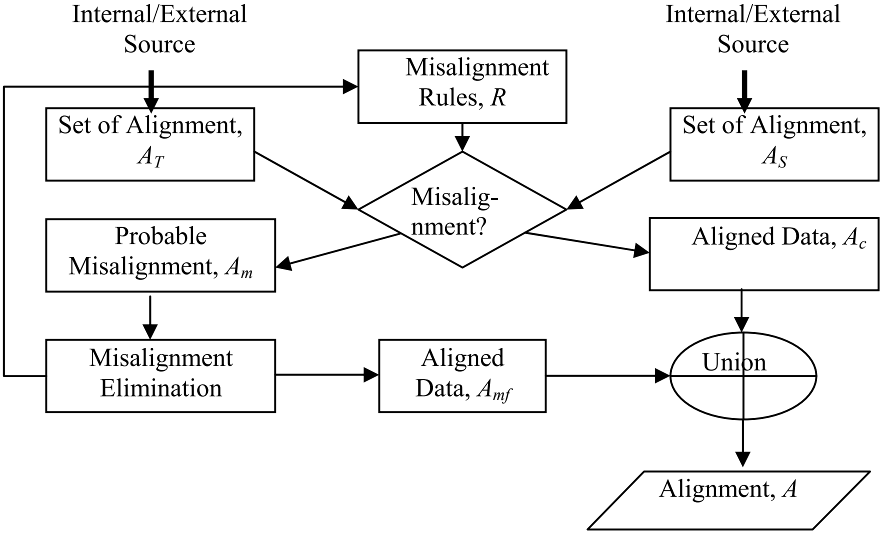


Fig. 1. Overall layout of misalignment module

3.1 Detection of Misalignment Candidates

Detection of basic misalignments is straightforward. As misalignment is defined above, it is detected by applying the set operations stated in equation (2).

The extended misalignment detection is given in the Algorithm 1 in Figure 2.

```
Algorithm 1. Extended-Misalignment-Detection

Input: Two ontology-alignment-sets $A_T$ and $A_S$, rule-set, R
Output: set of misalignment, $A_m$

$A_c = A_S \cap A_{T;}$
$A_m = (A_S \cup A_T) - A_{c;}$
for each element $a_i \in A_c$
    Let, $e_{o1}$, $e_{o2}$ be associated with $a_i$;
    Test the satisfiability applying rules, R;
    if not satisfied then
            $A_m = A_m \cup a_i$;
end for
output $A_m$;
```

Fig. 2. Algorithm of detecting extended misalignment applying rules

3.2 Elimination of Misalignment Candidates

Elimination of misalignments is a non-trivial task of calculating overheads. Two entities e_{o1} and e_{o2} are aligned as a_i, where a_i is a member of misalignment set A_m. Then similarity between e_{o1} and e_{o2} are recalculated. The recalculation processes are described below.

Terminological Similarity

Terminological similarity between e_{o1} and e_{o2} were recalculated using WordNet hyponym and hypernym or an *'is-a'* relation. It was calculated by measuring the distance of the two entities in WordNet where entities were organized into hyponymy/hypernymy taxonomies [14]. Su and Gulla [16] used a semantic distance measure to strengthen the mappings of instances whose concept names were closely related in WordNet; Silva and Rocha [15] also used a semantic distance measure, adapted from that proposed by Resnik in 1995 [14]. Similar approach was used in our system to calculate semantic distance e_{o1} and e_{o2}. Semantic similarity between the entities was the inversion of the distance.

Bottom-Up Similarity

One-to-one alignment between parents of entity, e_{o1} and parents of entity, e_{o2} were looked up from the existing alignment. The closer the alignment available, the higher the chances of alignment between e_{o1} and e_{o2} held. If there was no alignment between any parent elements of parent-set of entities e_{o1} and e_{o2}, not-aligned probability was considered higher.

Relation-Based Similarity

OWL ontology itself has some important relations. Some of them are positive to the similarity calculation while some of them have negative impact to calculate similarity. The positive-ness similarity and the negative-ness distance were calculated considering direct positive-ness tags or negative-ness tags and considering that of other associated entities as well.

Similarity Aggregation and Decision

The owl:equivalent, owl:disjointWith, owl:inverseOf and some other predefined relationships are considered having strong impact in the misalignment elimination process. Other similarity values like terminological-similarity, bottom-up similarity and other relation-based similarity, are considered with equal impact, i.e. associated with equal weights.

$$Sim_t = \sum_{i=0}^{n} w_i s_i \qquad (2)$$

where n is a finite predefined number of similarity, s_i is similarity value extracted from i-th similarity computation subsystem and w_i is the weight associated to s_i and we considered

$$w_1 = w_2 = \ldots\ldots\ldots = w_n \text{ and } \sum w_i = 1 .$$

If Sim_t > threshold, alignment between e_{o1} and e_{o2} holds; otherwise, misalignment is confirmed.

4 Conclusions

Our method of misalignment detection and elimination facilitate the ontology alignment system boosting up precision and recall. Intelligent autonomous agent of semantic web may take advantages using this misalignment detection and elimination process.

At present, misalignment detection and elimination process is applied in intra-ontology-alignment system of our OntoKDE. Our future plan is to evaluate this process feeding two sets of alignment retrieved from two different ontology alignment systems.

The basic rules of misalignment are retrieved manually by observing the structure of OWL ontology. Our future target is to enhance this module of misalignment detection and elimination by improving the knowledge of rules of misalignment using reinforcement learning.

References

1. Do, H.-H. and Rahm, E.: COMA - a system for flexible combination of schema matching approaches. In Proc. of the 28th VLDB Conference, pp 610–621, Hong Kong, Aug. 2001
2. Ehrig, M., Haase, P., Stojanovic, N. and Hefke, M.: Similarity for ontologies - A Comprehensive Framework, In 13th European Conference on Information Systems. May 2005
3. Euzenat, J., Stuckenschmidt, H. and Yatsevich, M.: Introduction to the Ontology Alignment Evaluation 2005 [online]. October, 2005 [cited 12 Jan 2006] Portable Document Format. Available at: <http://km.aifb.uni-karlsruhe.de/ws/intont2005/intontPresentationsOAEI.pdf
4. Euzenat J.: An API for ontology alignment. 3rd International Semantic Web Conference (ISWC), pp 698-712, Hiroshima, Japan, 2004
5. Euzenat, J. and Valtchev, P.: Similarity-based ontology alignment in OWL-lite. 16th European Conference on Artificial Intelligence (ECAI), Nov., 2004.
6. http://wordnet.princeton.edu/man/wnstats.7WN
7. Hu, W., Jian, N., Qu, Y. and Wang, Y.: GMO: A graph matching for ontologies, K-Cap Workshop on Integrating Ontologies, pp 41-48, Banff, Canada, 2005
8. Jian, N., Hu, W., Cheng, G. and Qu, Y.: FalconAO: Aligning ontologies with Falcon, K-Cap Workshop on Integrating Ontologies, pp 85-91, Banff, Canada, 2005
9. Langone, H., Haskell, B. R. and Miller, G. A. Annotating WordNet, Frontiers in Corpus Annotation 2004, HLT-NAACL Conference Workshop, pp. 63-69
10. Levenshtein, I. V.: Binary Codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory, 10(8): pp 707-710, 1966
11. Maedche, A. and Staab, S.: Measuring similarity between ontologies. Proc. of the EKAW, Springer LNCS, 2002: 251-263
12. Melnik, H., Garcia-Molina, and E. Rahm: Similarity flooding: A versatile graph matching algorithm. In Proceedings of the International Conference on Data Engineering (ICDE), pp 117–128, 2002

13. Porter M.F.: An Algorithm for Suffix Stripping, In: Sparck Jones, Karen, and Peter Willet (eds.), Readings in Information Retrieval, Morgan Kaufmann, San Francisco, 1997
14. Resnik, P. Using information content to evaluate semantic similarity in a taxonomy. In Proc. 14th IJCAI, pp 448–453, Montréal, Canada, 1995.
15. Silva, N. and Rocha, J. MAFRA – An ontology MApping FRAmework for the semantic web. In Proc. 6th International Conference on Business Information Systems, Colorado Springs, USA, 2003.
16. Su, X. and Gulla, J. A. Semantic enrichment for ontology mapping. In Proceedings of the 9th International Conference on Applications of Natural Language to Information Systems, *(NLDB'04),* pp. 217-228, Manchester, UK, 2004. Springer.

Semantic Integration of Enterprise Information: Challenges and Basic Principles

Jingtao Zhou and Mingwei Wang

The Key Laboratory of Contemporary Design and Integrated Manufacturing Technology,
Ministry of Education, Northwestern Polytechnical University, Xi'an, China, 710072
zhou.jingtao@gmail.com, Wangmv@nwpu.edu.cn

Abstract. To overcome the challenges of EII (Enterprise Information integration), we propose SGII which is the first system undertaking research at the intersection of semantic grid and P2P data integration, exploiting their strengths in a common framework, and expanding their applicability in the area of EII. We first discuss how the P2P and semantic grid technologies can drive current EII systems to a new decentralized, flexible, scalable system based on a short survey of the state of the art of EII and its current challenges. Then, through a discussion of the fundamental formal architecture in general and its components in particular, we depict the basic integration principles from both P2P and semantic grid perspectives. The key contributions of this paper are a P2P semantic grid service oriented framework for EII, which mainly consists of three semantic grid services (data peer, semantic peer and application peer services); basic integration principles, which is compatible with OGSA-DAI infrastructure and P2P data integration paradigm; and added value over the state of the art of EII.

1 Introduction

The global changes in economic, shift of the competitive edge and the advent of new technology continually alter the manufacturing and business environment in which enterprises operate. Manufacturing and business management has been extending outside an enterprise in a distributed form geographically or according to business logic spreading across multiple enterprises [1]. The trend of long- or short-time close collaboration across the whole vertical and horizontal manufacturing industries has been illustrated by the implementation of e-commerce, e-business, or virtual enterprise [2]. Surviving in such an increasing globalization and flexibility environment requires an extremely flexible, self-adaptive IT infrastructure capable of integrating and coordinating any involved information from any heterogeneous data sources, applications, and environments on demand to facilitate interoperation and collaboration over large-scale computer networks.

However, in spite of extensive R&D and successful pilots, traditional enterprise information infrastructure is poorly suited for dealing with the strategic, long-term barriers to efficient information sharing across enterprise internal and external boundaries over highly complex and dynamic networked environment. The lack of suitable basic framework leads to many information solutions that have to make overmuch tradeoff between long-term adaptability and short-term applicability, broad interop-

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 219 – 233, 2006.
© Springer-Verlag Berlin Heidelberg 2006

erability and tailored function for very specific purposes. They can not reap the full potential benefits of information, and ultimately fail to the pursuit of setting and realizing corporate strategic and tactical goals. One underlying problem has remained unsolved yet: data resides in thousands of incompatible formats and cannot be systematically and understandably managed, integrated, and reused. As a result, there is mounting pressure from enterprise itself and outside for a direct move away from disparate information systems operating in parallel towards a more common and fundamental shared architecture for information interoperation.

The remainder of the paper is organized as follows. In section 2, the challenges of EII are investigated based on a short survey of current approaches. Then, a new solution for EII based on P2P semantic grid is discussed in section 3. In section 4 and 5, the basic information integration principles of the new solution are presented from both P2P and semantic grid perspectives respectively. Section 6 gives the concluding remarks.

2 Challenges of Enterprise Information Integration

2.1 Survey

Enterprises have long recognized the value of data integration. Efforts can be roughly classified into two categories: application centric integration (ACI) and data centric integration (DCI).

ACI, such as point to point integration and enterprise application integration (EAI), integrates relative data by linking applications through custom-coding or integration broker that acts as a hub to route messages between connected applications. Connection at the programmatic level, difficulty of metadata reuse, N^2 problem at data layer, multiple vendors for multiple systems, lack of common protocols, and tight-coupling of technology and systems in the end, make these solutions difficultly suitable to an open, dynamic information interoperation environment of businesses and operations in or across enterprises. In fact, ACI provides little data integration because it operates at the business-process level rather than data level.

DCI can be implemented by creating either a centralized repository for data access and analysis, such as data warehouses, or a data integrating layer over a set of distinct and autonomous data sources, such as federated information systems. Data warehouses fall short where an enterprise needs to address one need today and incrementally add additional requirements over time, or where there is a great deal of change in business requirements[3]. Federated information integration approaches can be divided into two classes, including tightly-coupled and loosely-coupled systems. Tightly-coupled system integrates all data sources by creating logical mappings between them and a single global schema (a single model or ontology) which may be derived from the actual data sources themselves, or the modeling of current and even future business and operation aspects of information. The main drawbacks when using a global schema are the difficulty of creating a single sufficient global schema to represent a large-scale data sources, maintain of changing and evolution caused both from data sources and the global schema itself. Hence, tightly-coupled approach is only adapted to small-scale integration with little independently changing and evolving. By contrast, the loosely-coupled approach coordinates autonomous component data sources without a single global schema, instead, with a set of federated schemas. In the context of real-time information integration, loosely-coupled federation system is more effective and has advantages than other approaches. However, early loosely-coupled federated systems are not broadly applied in real enterprise environment because of the using of private protocol and data model, low performance, laborious process, critical implementation conditions, immature technology and the lack of reliable infrastructure. Several key problems such as discovery of relevant data, semantic conflicts and violation between the autonomy and privacy hinder their further application, too.

With recent advent of new technologies such as web service, XML and SOA, and new drivers behind e-commerce, e-business and e-enterprise, a representative approach for enterprise information integration (EII) is proposed, which intends to integrate any form data in enterprises, and aims to provide a uniform interface for information access, manipulate, and integrate across multiple data sources. Renaissance of data integration begins with the forming of EII industry. Broadly speaking, the architectures underlying most current new EII approaches (e.g. IBM DB2 II[4], BEA Liquid [5], MetaMatrix [6] ,etc.) are still based on similar principles of loosely-coupled federated systems although they may support broad type data sources, use new XML model, speak with common protocols and publish integrated results with web services. Therefore, some traditional problems are inherited from loosely-coupled federated information integration system, such as scalable and semantic problem. Furthermore, current EII solutions may encounter their own fierce challenges when they need to integrate all involved information across the whole vertical and horizontal management logic, from both intra-enterprise and inter-enterprise on a highly complex and dynamic networked environment in a timely fashion. The circumstances seem to be too rigorous but are actual in real-world manufacturing and business environment.

2.2 Challenges

A resent survey [7] has addressed four challenges of EII including scaleup/performance, horizontal or vertical growth, integration with EAI, and metadata management/ semantic heterogeneity from a more general point of view. In this section, we will take a further discussion from a technical perspective.

- Scalability. The framework of most current EII systems is constructed as a hierarchy framework, in general, with (fix) multi-layer, which can be considered as a variation of the traditional five-level federated architecture [8], and may consist of local schema, export schema (possible wrapped by web service), federated or mediator schema (logic data view), and extern schema (data view for specific application and user, possible wrapped by web service). Generally, approaches relying on a priori creation of federated views do not scale-up efficiently given the complexity involved in constructing and maintaining a shared schema for a large number of possibly independently managed and evolving, sources[9]. A solution to this problem is to represent federated schemas by a hierarchy of small finely granular schemas, such as business entity schema used in [6] [10]. However, this approach needs more endeavors from specialists and relies on their strong intimate knowledge of the desired business entities to be created and deep understanding of the data, underlying schema, and relationships across the various data sources (e.g. [6] [10]), which becomes a drawback for scalability when this knowledge grows and changes as more sources join the system and when sources are changing [9].

- Horizontal and Vertical Integration. From the view of business and manufacturing, data integration requirements come from both the vertical and horizontal logic level. The framework of most current EII system is more appropriate for horizontal dimension of data integration rather than vertical dimension. Although the using of web service in some systems (e.g. data services in BEA liquid [5][10]) has shifted the focus on vertical data integration by service composition, the connection of vertical and horizontal integration has been artificially dissevered by the isolated definition of business concepts, which are dynamically composed only according to the specific application logic, only involving sources that have direct complementary data described by the corresponding federated schema, and ignoring the nature semantic relationship between these schemas indirectly. As a consequence, it can not consider the potential relative data in other data sources even though the relationship is implied by other federated schema which is not directly involved in an integration process. It does not consider the probable incompleteness of some sources, either. In fact, these scenarios can be avoided by establishing the network of relationships among federated schemas as well as data sources to connect the horizontal and vertical data integration.
- Centralized Integration. Most implementations of EII generally resolve queries from dedicated servers that house federated metadata. Not only do these dedicated servers generally form a bottleneck in terms of scaling performance, but the centralized computing model can be a scalability bottleneck in terms of administration [11]. Federated schema design must be done globally; any changes of the federated schema can be only made by the central administrator. This can be especially challenging when data is owned and managed by numerous heterogeneous groups with different needs, and when integrate data across organizations.
- Semantic. Even for information that is carried by web service and represented using XML, a serious and expensive barrier to dramatic improvement exists: a severe lack of explicit knowledge about associated corporate information. The thorny question of locating and understanding the data to be integrated still remains [7].To fill this gap, an integration framework enabling the semantic interoperability across an enterprise even virtual enterprises both at data level and service level is needed.

3 EII on Top of P2P Semantic Grid

Clearly, to overcome the challenges of EII and create a more flexibly and dynamically semantic interoperation environment for enterprise information over a large-scale computer networks, we believe there is a need for a new class of data sharing infrastructure. Such infrastructure is fundamentally distributed and dynamic networked, supports highly flexible sharing relationships, ranging from client-server to peer-to-peer, addresses scalability as well as resource control, and achieves interoperability at not only system level but semantic or knowledge level.

Fortunately, emerging new technologies including Grid [12], P2P [13], and Semantic Web [14] explore the requirements and approaches in the above context to some extent, which are all concerned with the organization of resource sharing in large-scale societies.

3.1 Data Integration Based on Grid

In the context of information integration and sharing, grid technologies distinguish current information integration technologies in enterprise by providing not a generic approach but also an open and standard-based infrastructure. The current efforts of the data grid community concentrate on providing a global, uniform access methodology for all database resources. Meanwhile, information grid projects shift the emphasis on information integration and mediation. However, in spite of data grid, information grid or Grid-based Virtual Databases [15], they are based on the similar principles of loosely-coupled federated system and faced the same challenges of current EII systems when apply them in EII environment. This situation will be changed when grid is extended to semantic grid, which wants to create an internetcentered interconnection environment on the grid to effectively organize, share, cluster, fuse, and manage globally distributed versatile resources based on the interconnection semantics [16]. By reflecting principles of Semantic Web and Semantic Web service in grid environment, semantic grid enables not only standardized loosely-coupled interoperability but also semantic or knowledge level information coordination. However, although semantic grid could provide an appropriate integration and operation infrastructure for enterprise information as discussed in our previous SGII approach [17], whether it is really adapted to dynamic enterprise integration environment that still requires substantial research because it is presently immature and requires solutions to issues of self-adaptation, fault tolerance, and scalability. Fortunately, P2P technology has much to offer for this scenario.

3.2 P2P Data Integration

Differently from popular data integration approach in enterprise, peer-to-peer data integration architecture achieves information integration by establishing peer-to-peer mappings rather than using centralized schema. As a natural extension of semantic data integration approaches [18], P2P data integration framework has proved its abilities of dealing with intermittent data source participation and highly variable interoperation behavior. However, although P2P data integration attempts to avoid the using of a single global schema, some systems still need a global ontology or vocabulary, such as the global RDF ontology in [19]. In essence, in spite of the using of single ontology or the same domain assumption in [20], it implies that all peers belong to the same domain and makes these solutions unrealistic for EII. Furthermore, P2P developers have worked mainly on vertically integrated applications [21], failed to define common protocols, standardized infrastructures for interoperability and enough control of data sources, for which Grid may have more to provide.

3.3 Data Integration on P2P Semantic Grid

Therefore, neither semantic grid nor P2P data integration by itself is sufficient to be a competent architecture for EII, we believe such a complex application need the convergence of peer-to-peer, Grid and semantic (or semantic grid) computing as Foster has stated [22]. Undertaking the intersection fulfillment of these technologies in information sharing area will address scalability, semantic interoperability, selfadaptation, and failure recovery, while, at the same time, providing a persistent and standardized infrastructure for interoperability [21]. In this context, we extend our previous work SGII, which takes semantic-grid-based framework as foundation, to a p2p-semantic-grid enabled information sharing infrastructure by reflecting principles of P2P data integration in semantic grid environment. The new SGII is a fundamentally decentralized architecture both for data sharing and administration, which attempts to achieve scalability by realizing semantic information interoperation in a P2P way, overcome the semantic problem by replaces global federated schemas with a semantic grid services interconnection environment, and combine the vertical and horizontal integration logic by implementing the P2P integration paradigm on standardized Grid infrastructure.

4 Integration Principles from P2P Perspective

From the P2P point of view, SGII is a P2P information integration system *I* composed of a set of instances of three kinds of peers: Data Peer (*DP*), Semantic Peer (*SP*) and Application Peer (*AP*), each of which has a schema describing the data held by the

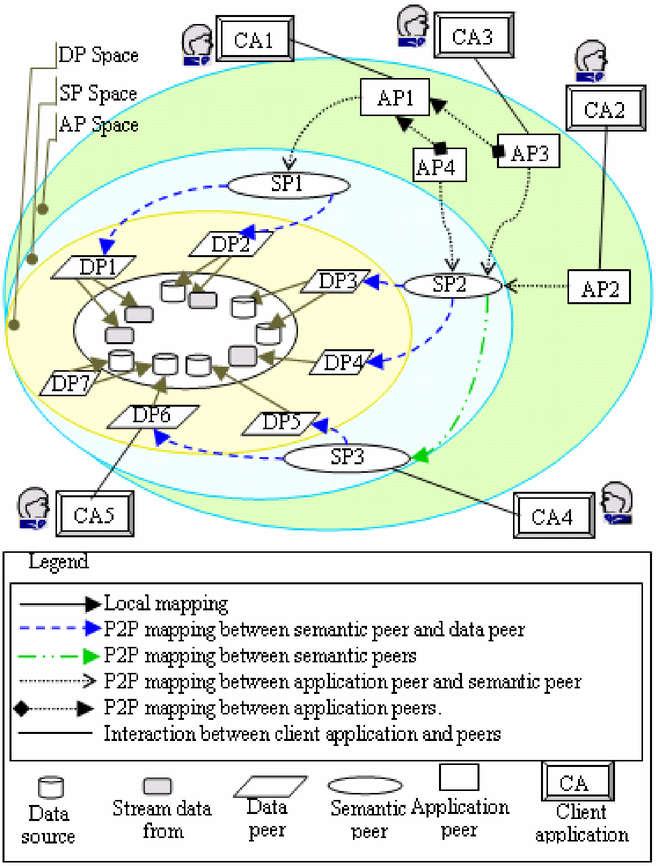

DP Space
SP Space
AP Space
CA1
CA2
CA3
CA4
CA5
AP1
AP2
AP3
AP4
SP1
SP2
SP3
DP1
DP2
DP3
DP4
DP5
DP6
DP7
Legend
Local mapping
P2P mapping between semantic peer and data peer
P2P mapping between semantic peers
P2P mapping between application peer and semantic peer
P2P mapping between application peers.
Interaction between client application and peers
Data source
Stream data from
Data peer
Semantic peer
Application peer
CA
Client application

Fig. 1. Framework of SGII

peer, and a set of P2P mappings that specify the semantic relationships with the data exported by other peers. As shown in figure 1, instances of each kind of peer construct corresponding spaces called *DP* Space, *SP* Space and *AP* Space respectively. The formal framework of SGII is $I = \{\{DP_1, DP_2, \ldots, DP_m\}, \{SP_1, SP_2, \ldots, SP_n\}, \{AP_1, AP_2, \ldots, AP_k\}\}$, where m, n, and k are the quantities of corresponding *DP*, *SP* and *AP* nodes. Logically, this formal framework is an extension of the logic models of P2P data integration proposed in [23][24].

Intuitively, SGII supports any arbitrary network of relationships between peers, but in fact, mappings between peers will be restricted according to the actual semantic relationships among peers' schemas and the type of peers. Note that the decentralized P2P data integration paradigm allows any user to contribute new data source, peers, or even mappings between peers locally or globally, not always globally as current EII system do. Therefore, theoretically, SGII allows any component, in spite of data source or peer, to have its own autonomous administration domain, which avoids centralized integration naturally. In practice, data sources and peers may be controlled by communities in an enterprise or across several enterprises in the context of EII.

4.1 Data Peer

Each *DP* behaves both as a control point and a mediator of a set of autonomous data sources, which is defined as a tuple $DP = (D, L, M_{DL}, M_{DS})$, where

- *D* is the peer schema held by *DP* to represent the intensional description of the data controlled by *DP*. *D* can be regarded as an export schema for local schemas *L* from data sources.
- *L* is a set of local schemas of data sources, which represents the data exposed (shared) by data sources.
- M_{DL} is a mapping between *D* and *L*, which consists of a set of assertions establishing the connection between the elements of *D* and those of *L*. The mapping M_{DL} follows the GLAV (Global-Local-As-View) [25]paradigm, which means that conjunctive queries of $Q(D)$ and $Q(L)$ with the same arity have the relation $Q(L) \subseteq Q(D)$, i.e. every source relation is defined over the data peer schema. The definitions of *D*, *L*, M_{DL} are similar to the definitions of *peer schema*, *source schema* and *local mapping* in [23].The motivation to use exactly the GLAV approach is mainly due to its ability of easily sources adding and removing, i.e. the scalability for the dynamic connection of data sources with a *DP*. In the end, it will contribute to the whole scalability of *DP* space. Usually, data sources may belong to a community, a sub-community, or even an authorized individual, as a consequence the corresponding *DP* often belong to the corresponding owner.
- M_{DS} is a set of P2P mappings between *D* and the peer schema *S* held by *SP* to connect *DP* and *SP* following the LAV mapping paradigm[25], which means that conjunctive queries of $Q(D)$ and $Q(S)$ with the same arity have the relation $Q(D) \subseteq Q(S)$. Note that, in order to achieve a balance between the efficiency of centralized semantic mediation and distributed query, load balancing and robustness, the P2P mappings between *DP*s and those between *DP* and *AP* provided in previous versions of SGII [26] will not be supported in this version.

4.2 Semantic Peer

Each semantic peer *SP* is defined as a tuple *SP*= (*S*, M_{ss}), where

- *S* is the peer schema or ontology held by *SP* to represent the intensional knowledge of a specific domain in enterprise. For most enterprises, there usually exist some well-defined set of meta-data standards for specific domains, which can be used as initiate part of *S*. On the other hand, since P2P paradigm in SGII allows any user to contribute *SP* locally, *S* could be any granularity (any fine or coarse granularity) description for a domain, which completely lies on the provider's aim and decision. By *S* and M_{DS}, each *SP* works as a mediator for a group of *DP*s. We use *S* as the common semantic model or ontology agreement for a (sub-) community. Therefore, *SP* and relative *DP*s usually belong to the same (sub-) community in logic. Since *S* could be looked as an intensional knowledge of a specific domain, all the P2P mappings M_{DS} relative to *S* form part of extensional knowledge for the corresponding domain.
- M_{ss} is a set of P2P mapping assertions that express semantic relationships between two or more elements of ontologies or peer schemas of *SP*s. Therefore, the P2P network formed by connected *SP*s creates an alignment between different *SP*s' schemas, which is actually used to represent the intersection and semantic relationships among ontologies of different (sub-) communities. As *S* in *SP* is local rather than global, i.e., each *S* may be created by a local community which publishes its data sources with corresponding ontology or *S*. Global ontology, instead, is replaced with an interlinked collection of semantic mappings between semantic peers' individual schemas. This is more pragmatic than creating a global one or assuming all peers belong to the same domain like [20]. Mappings in M_{ss} are created following both GAV (global as view)[25] and LAV ways because of the difficulties to decide whether one *S* is more general than others. Furthermore, semantic conflicts between data peer schemas can be resolved by the mapping to common *SP* schema.

4.3 Application Peer

AP peers often work as delegations of actual enterprise-wide client applications (CA) which are directed to other type peers. Each application peer *AP* is defined as a tuple *AP*= (*A*, M_{AS}, M_{AA}), where

- *A* is the peer schema held by *AP* to define the enterprise-wide consistent unified representation of unit information model to serve the enterprise-level information requirements better, such as unit business entities like Customer, Product, etc. Since *A* tries to create a single view of information entities across the enterprise for enterprise-wide information requirements, it may be created by data consumers or business architects not data source providers to tailor information in a way that makes sense to particular applications or types of applications; it might have to be adjusted to represent the nuances of individual information requirement units. In practice, we define each *AP*'s schema *A* as a fine-grain unit model to respond to different information requirements agilely by orchestration relative *A*s in *AP*s.
- M_{AS} is a set of P2P mappings between *A* in *AP* and *S* in *SP*, which describes the semantic relationships between the domain model *S* referenced by the source data through M_{DS} and the unit information model *A* referenced by the information requirement definition from data consumer. Hence, networked *SP*s establish an ab-

stract semantic mediation layer both for source data and enterprise-wide information requirements. Intuitively, M_{AS} supports multiple-to-multiple mappings between *A* and *S*. However, in practice, we will break each *A* into a set of A's until the mapping from each A' to *S* is a one-to-one mapping. In this scenario, *A* establishes indirect mapping to *S* through the sub-set of A's. Therefore, each mapping in M_{AS} is defined following a GAV (global as view) approach, i.e. the conjunctive queries of $Q(A)$ and $Q(S)$ with the same arity have the relation $Q(A) \subseteq Q(S)$. Furthermore, the fine-grain unit model defined by each *AP*'s schema makes it agile to respond to different information requirements by orchestration relative unit models.

- M_{AA} defines a set of P2P mappings between *A*s residing in corresponding *AP*s to model the actual complex information requirements by orchestrating *A*s, i.e. the information units. By analogy with M_{SS}, mappings in M_{AA} are created following either GAV or LAV approach. From the modeling point of view, peer schema of *SP* could be regarded as domain ontology [27] in any granularity; a set of *A*s orchestrated through M_{AA} could be regarded as a modeling of a task ontology [27]. Therefore, the connected *SP*s network establishes an alignment of domain ontologies; the connected *AP*s network establishes an alignment of the components of task ontologies or task ontologies themselves. This makes P2P mappings in SGII be extremely complex than the solutions that take strict assumption of all peers belongs to one domain [20]. But in contrast, SGII has strong applicability than those solutions as well as guarantee the personalization of CA.

4.4 Semantic Mediation

Intuitively, the semantic mediation in SGII might be roughly classified into four categories: *DP-SP*, *SP-SP*, *AP-AP* and *SP-AP* mediations as shown in figure 2 from a) to d) respectively because of the existence of different types of peers and P2P mappings. However, all these mediations are based on the same fundamental: creating alignment of peer schemas by establishing P2P mappings among these nodes. Therefore, the only ways to unify and simplify these types of mediations is to use the same technology and language to create and describe all peers' schemas and P2P mappings. In this context, we use WSML [28] for expressing both peer schemas and P2P mappings rather than RDF(S) in previous version SGII [26], WSMT [29] as the corresponding tool to edit these schemas and mappings. The mediation is created in design-time, which is the basis for responding to the queries in SGII.

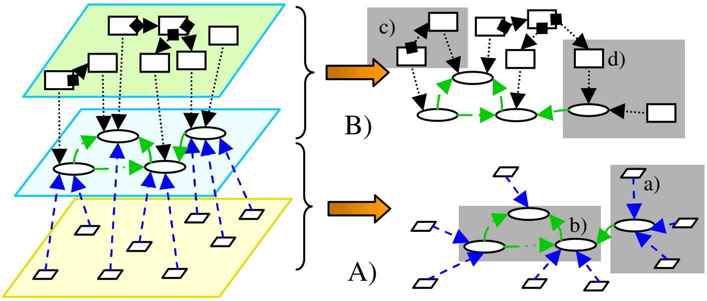


Fig. 2. Framework of SGII

4.5 Query

SGII supports three kinds of queries.

- D-Query, query posed on the peer schema *D* of *DP*. Locally, the query will be decomposed into sub-queries on relative data sources controlled by this *DP* according to the M_{DL} mapping to get answer from local sources. This local query process is called D-L-Query. Meanwhile, the source query is reformulated into a target query over the connected *SP* through *DP-SP* semantic mediation process based on the mapping M_{DS}. This query transfer process is called D-S-Query. The D-S-Query will trigger another kind of query, S-Query.
- S-Query, query posed on the peer schema *S* of *SP*. S-Query only involves the relative *DP*s and *SP*s in SGII, except *AP*s. In this context, the P2P network composed of *DP*s and *SP*s can be looked as a super-peer network shown in figure2 A), where *SP* is super peer and S-Query is the query on super peer. First, the S-Query is rewritten into sub-queries on *DP*s (called S-D Query) through M_{DS}, and sub-queries on neighboring *SP*s (called S-S Query) through *SP* semantic mediation process based on M_{SS}. Then, the S-D Query will trigger the D-L-Query process in each relative *DP* to get answer from data sources while the S-S Query will start new S-Query process (es) on the corresponding *SP*(s). In this way, the reformulations of S-Query will produce a directed-graph topology. By now, SGII does not support cyclic mappings between peers hence the reformulation directed-graph is tailored to a reformulation tree based on the same fundamental discussed in [24]. The reformulation process terminates when no S-S Queries are created. Note that if the S-Query is triggered by a D-S-Query, the *SP* will not reformulate the sub-query on the originating *DP*.
- A-Query, query posed on the peer schema *A* of *AP*. A-Query is reformulated into sub-queries on *SP* (called A-S Query) through *SP-AP* mediation process based on M_{AS}, sub-queries on neighboring *AP*s(called A-A Query) through *AP-AP* mediation process based on M_{AA}. The A-S Query will trigger the S-Query process (es) while the A-A Query will start new A-Query process(es) on the corresponding *AP*(s). By analogy with S-Query, A-Query produces a tree reformulation, too.

Because the Peer schemas and P2P mappings are represented in WSML, we use WSML2Reasoner framework [30] with KAON2 reasoner for reasoning and querying with WSML descriptions, which can deal with a large subset of WSML-DL, namely *SHIQ*(D)[31]. All the queries are represented as conjunctive queries. The queries in DL will be translated into conjunctive queries in datalog. Using conjunctive queries in DL has been well studied in [32].

5 Integration Principles from Semantic Grid Perspective

From the grid point of view, SGII is a semantic grid service oriented architecture, i.e., every peer in SGII is independently realized as a grid service based on the Open Grid Service Infrastructure and will be a semantic grid service by semantic enrichment. In particular, each peer service, by now, is designed as OGSA-DAI grid data service (GDS) [33] compatible by extending the GDS port types of OGSA-DAI, and enriched by describing the corresponding P2P mapping relations using the metadata (represented using WSML) of GDS. Besides data access, OGSA-DAI enables some data integration functions by providing facilities for combining or transforming data from multiple data access components through the document-oriented interface of GDS. As a result, each P2P operation will be described as the *activity* of *perform documents* [33] of GDS and support complex P2P interactions by the composition of activities.

5.1 Semantic Peer Services

As a GDS, each peer service in SGII implements the port types GDS and GDT (Grid Data Transport) from OGSA-DAI, and the Grid Service port type from OGSA. Other port types from OGSA, such as NSnk (*NotificationSink*) and NSrc (*Notification-Source*) can also be optionally implemented according to the requirements. Each peer's schema, relative P2P mappings and data service descriptions are described in the metadata (represented using WSML) of corresponding peer service.

Therefore, we alter our previous definition of peer services in [26] as follows:

- The Mapping (M) port type, which defines a facility for the P2P mapping between peers by interacting with the interfaces and tools provided by WSMT. Previous GAV/LAV mapping port types [26] are replaced by unified Mapping port type.
- The GetMapping (GM) port type, which is used to get the P2P mapping metadata from Peers.
- According to the different kinds of queries in SGII, each kind of peer defines its own Query (Q) port type rather than GAV/LAV Query port types defined in [26] to perform queries posed on it. The Query port type accepts conjunctive queries as input.

In the context of data service description, the semantics of peer schema is a more powerful criterion than the similar port types, so we implement peer service as semantic grid service by enriching it using its peer schema itself. We can discover the peer service through the query posing on relative peer schemas, such as find the *DP* service by the query on *SP* schema, which may have direct or transferable connection with the *DP* service expressed by P2P mappings.

5.2 Query Implementing Based on Peer Services

Since each peer service is a GDS, its instances are created based on the same mechanism of GDS instances by invoking the GDSF (Grid Data Service Factory)[33],

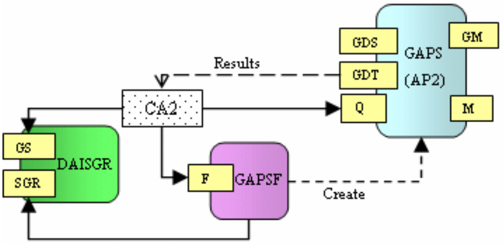


Fig. 3. Creation of *AP* instance

whereas the DAISGR (*Database Access and Integration Service Group Registry*)[33] allows clients to search for GDSs and GDSFs. Here, the clients can be either CAs (e.g. CA2 in figure 3) or peers that need invoke sub-peers (cf. *AP*2 in figure 4 and 5). In the context of query, it means that the originating peer is responsible for the instantiation of neighboring peers involved in the query process. Therefore, from the service point of view, a query process in SGII is a process of peer services instantiation and interaction through the Query port types along with the query reformulation. In brief, the instantiation mechanism of peer service provides an easy way to implement query process based on services for SGII. We take A-Query posed on *AP*2 by CA2 in figure1 as example, part interactions between peer services are shown in figure 5.

In figure 3, figure 4 and figure 5, GAPSF (Grid AP Service Factory), GDPSF (Grid DP Service Factory), and GSPSF (Grid SP Service Factory) are service factory services for *AP*, *DP* and *SP* respectively, which all implement the port types of GDSF; GAPS (Grid AP Service), GDPS (Grid DP Service) and GSPS (Grid SP Service) are peer services of *AP*, *DP* and *SP* respectively, which all implement the GDS port type.

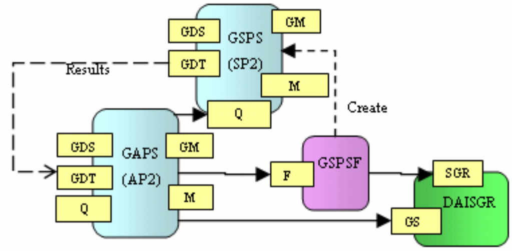


Fig. 4. Creation of sub-peer instances

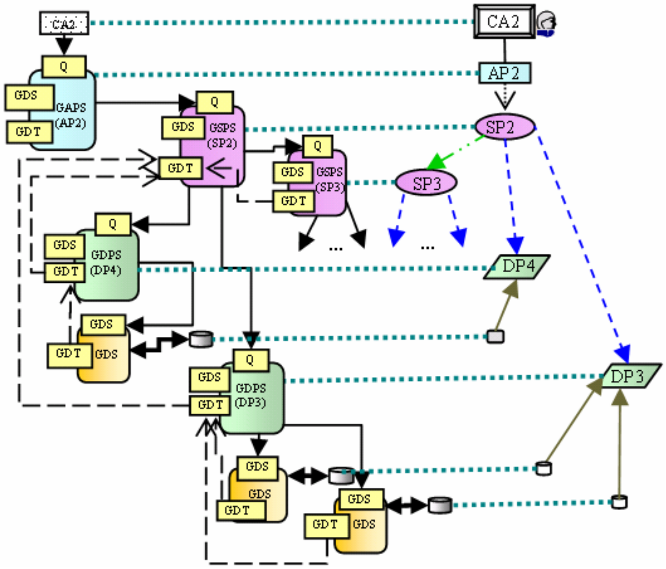


Fig. 5. Query implementation based on peer services interaction

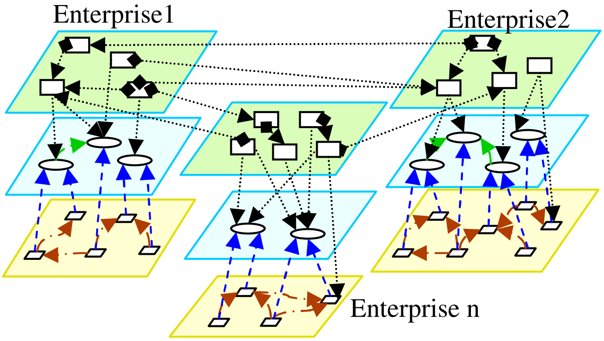


Fig. 6. Information integration across enterprises

SGII also allows the query interactions among *AP*s residing in different enterprises to support the information integration across enterprises, such as virtual enterprise. In this context, queries among *AP*s are similar to the business level interoperation. This scenario can be shown as figure 6.

6 Conclusions and Acknowledgements

As a specific issue of data integration, integrating enterprise information across enterprises is a challenge to both traditional and new data integration approaches. The application based on Semantic Grid and P2P frameworks for provision of effective information resource sharing in and across enterprises is promising. However it needs further researches and industrial case studies to be carried out in order to evaluate finally the utility of information and semantic interoperability based on SGII.

The authors would like to thank to Jos de Bruijn for his advice on how to use WSML relative technologies in the data integration environment, Adrian Mocan and Mick Kerrigan for the discussion about ontology mediation technologies and WSMT in WSMX.

References

1. Wang, Q., Yung, K. L., Hung, Wai.: A hierarchical multi-view modeling for Networked Joint Manufacturing System. Computers in Industry 53 (2004) 59–73
2. Nahm, Y.-E., Ishikawa, H.: A hybrid multi-agent system architecture for enterprise integration using computer networks. Robotics and Computer-Integrated Manufacturing, Vol.21 (2005) 217-234
3. Nimble Technology: Next-Generation Data Integration: Harnessing Data for Business Advantage (2002)
4. Bruni, P., Arnaudies, F., Bennett, A., et al. : Data Federation with IBM DB2 Information Integrator V8.1. IBM Redbook (2003)

5. Liquid Data Engineering Team: Liquid Data for WebLogic: Integrating Enterprise Data and Services. ACM SIGMOD 2004 (2004)
6. Hauch, R., Miller, A., Cardwell, R.: Information Intelligence: Metadata for Information Discovery, Access, and Integration. ACM SIGMOD 2005 (2005) 793–798
7. Halevy, A. Y., Ashishy N., Bitton D., et al.: Enterprise Information Integration: Successes, Challenges and Controversies. ACM SIGMOD 2005 (2005)778–787
8. Sheth, A.P., Larson, J.A.: Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys, Vol. 22, No. 3(1990)183–236
9. Choucri, N., Madnick, S. E., Moulton, A., et al.: Information Integration for Counter Terrorism Activities: The Requirement for Context Mediation. Working Paper CISL# 2003-09(2003)
10. BEA: Services Platform Concepts Guide, Version: 2.0.1(2005)
11. Ives, Z. G.: Efficient Query Processing for Data Integration. Phd thesis, university of Washington (2002)
12. Foster, I., Kesselman, C. (eds): The Grid: Blueprint for a New Computing Infrastructure. San Francisco, CA: Morgan Kaufmann Publishers (1998)
13. Oram, A. (ed.): Peer-to-Peer: Harnessing the Power of Disruptive Technologies. Sebastapol, California: O’Reilly (2001).
14. Berners-Lee, T., Hendler, J. and Lassila, O.: The semantic web. Scientific American, Vol. 284, No.5, (2001) 34–43
15. Paton, N., Atkinson, M., Dialani, V., et al.: Database access and integration services on the grid. Technical Report UKeS-2002-03, UKe-Science Programme(2002)
16. Zhuge, H.: Semantic Grid: Scientific Issues, Infrastructure, and Methodology. Communications of the ACM, Vol. 48, No. 4 (2005) 117–119
17. Zhou, J.T., Zhang, S.S., Zhao, H., Wang, M.W.: SGII: Towards Semantic Grid-based Enterprise Information Integration. LNCS 3795 (2005)560–565.
18. Ruzzi, M.: Data Integration: state of the art, new issues and research plan(2004)
19. Cruz, I. F., Xiao, H., Hsu, F.: Peer-to-Peer Semantic Integration of XML and RDF Data Sources. *AP*2PC 2004 (2004)
20. Calvanese, D., De Giacomo, G., Lenzerini, M., et al.: Hyper: a framework for peer-to-peer data integration on grids. In: ACM SIGACT SIGMOD SIGART PODS 2004 (2004)
21. Androutsellis-Theotokis S., Spinellis D.: A Survey of Peer-to-Peer Content Distribution Technologies. ACM Computing Surveys, Vol. 36, No. 4 (2004) 335–371
22. Foster, I., Iamnitchi, A.: On death, taxes, and the convergence of peer-to-peer and grid computing. In IPTPS’03, Berkley, CA (2003)
23. Calvanese, D., Giacomo, G.D., Lenzerini, M., Rosati, R.: Logical Foundations of Peer-To-Peer Data Integration. PODS 2004 (2004) 241-251
24. Halevy, A.Y., Ives, Z.G., Suciu, D., Tatarinov, I.: Schema Mediation in Peer Data Management Systems. ICDE'03(2003) 505
25. Lenzerini, M.: Data Integration: A Theoretical Perspective. In: Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on PODS(2002) 233–246
26. Zhou, J.T., Wang, M.W.: SGII: Combining P2P Data Integration Paradigm and Semantic Web Technology On Top Of OGSA-DAI. CCGRID 2006(2006) 93-98
27. Perez, A.G., Benjamins, V. R.: Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods. In: proceedings of the IJCAI-99 (1999)
28. de Bruijn, J., Lausen, H., Krummenacher, R., et al.: The Web Service Modeling Language WSML, http://www.wsmo.org/TR/d16/d16.1/v0.2/20050320/

29. Kerrigan, M.: Web Services Modeling Toolkit (WSMT), http://www.wsmo.org/TR/d9/d9.1/
30. http://dev1.deri.at/wsml2reasoner/
31. de Bruijn, J., Feier, C., Keller, U., et al.: WSML Reasoning Implementation, http://www.wsmo.org/2004/d16/d16.2/v0.2/20041220/
32. Calvanese, D., De Giacomo, G., Lenzerini, M.: Answering queries using views over description logics knowledge bases, In Proc. of the 17th Nat. Conf. AAAI(2000) 386–391
33. Antonioletti, M., Jackson M., Krause, A., et al.: The design and implementation of Grid database services in OGSA-DAI: Research Articles. Concurrency and Computation: Practice & Experience Vol. 17, Issue 2-4 (2005)357 – 376

Application Integration Using Conceptual Spaces (CSpaces)

Francisco Martín-Recuerda

Digital Enterprise Research Institute (DERI), Leopold-Franzens Universität Innsbruck,
6020, Austria
francisco.martin-recuerda@deri.org

Abstract. Application integration is a complex problem that consumes a significant share of the IT budget of many companies and organizations. *CSpaces* aim to improve the current state of the art in system integration by transforming the Semantic Web into a *Semantic Enterprise Service Bus* for application integration and coordination. A use case scenario for integration of heterogeneous project management applications in the construction industry tests the ability of CSpaces to handle integration problems.

1 Introduction

It is well known that companies spend a significant share of their overall IT budget on solving integration problems: Gartner Group estimates in a recent study that around 40% of the overall IT budgets of companies is used just for system integration efforts. The integration of software applications is often complicated by the diverse data structures and formats employed, the variety of communication protocols used, and the different interaction patterns implemented. Software vendors, academia and standard organizations have spent more than 40 years looking for the *holy grail* of system integration.

W3C is promoting the use of ontologies for annotating Web Services in order to improve automation of software integration[1]. [1] pointed out that the Semantic Web has the potential to become in an infrastructure for application integration and coordination. Fensel [1] promotes the idea of the transformation of the Semantic Web into a persistent shared memory for Semantic Web services. **Triple Space Computing** [1] combines a simple and powerful coordination model for asynchronous communication, tuplespace computing [2], with all the theoretical benefits of machine processable semantics for application integration. The attractive proposal of Fensel needs to reconsider some design principles to reach its full potential. Firstly, [1] promotes *REST* (Representational State Transfer) [3] as an abstract model for Triple Space Computing, but according to [4], REST is not appropriate for the implementation of an asynchronous communication infrastructure. Secondly, the tuplespace coordination model provides time and space decoupling but not *flow decoupling* [5] from the client side. Thirdly, Fensel does not address the problem of using *heterogeneous ontologies* for representing the data that Semantic Web Services aim to publish and read.

[1] http://www.w3.org/2002/ws/swsig/

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 234 – 248, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Fourthly and final, Triple Space Computing relies on the current status of the *Semantic Web* proposal in which several relevant questions are still open: how to keep coherence and consistency between the Web (including Web services descriptions) and the semantic annotations and how to annotate web pages that are dynamically generated (*dichotomy problem*); how to store and reason with the huge amount of semantic annotations that the Semantic Web will require (*scalability problem*); how to organize and share semantic annotations and how to persuade current web users to participate in the creation process of machine processable semantics (*publishing problem*); how to overcome conflicting terminology and conceptualizations defined by different ontologies (*heterogeneity problem*); how to ensure meaningful answers from reasoning engines when the information stored is not consistent (*inconsistency problem*); how to guarantee that only a restricted amount of users can visualize and edit concrete semantic annotations (*security problem*); and how to guarantee validity and trustworthiness of the semantic annotations (*trust problem*). In addition, a reference architecture specification for the Semantic Web is still missing, and after five years we have yet to rely on the *Semantic Web layer cake*[2].

In particular, how to handle *heterogeneity* and *trust* in the Semantic Web is very important for application integration. The Semantic Web is a distributed and open system in which heterogeneity cannot be avoided and trustworthiness cannot be guarantee. Regarding the *heterogeneity* problem, it is expected that dealing with many different heterogeneous ontologies with overlapping domains [6] will be the most likely Semantic Web scenario. Ontology merging and mapping are the main approaches that have been identified for dealing with heterogeneous ontologies. In the former approach, applications that rely on specific ontologies, can become inoperative after a merging process because the source ontologies are eliminated [6]. On the other hand, scalability is the main limitation of the point-to-point mapping approach, because it requires $O(n^2)$ ontology mappings, where n is the number of ontologies [7]. Several alternatives have been proposed to overcome the limitations of the merging and mapping approaches. For instance, the authors of the *PSL Ontology* [7] promote the use of heavy ontologies as an interlingua for ontology and application integration instead of using point-to-point translators. The conclusions of the study presented on [6] suggest a scenario of networks of ontologies organized around influenced ontologies that the authors called *ontology islands.* [6] extends and complemented the proposal of [7] and represents a sound proposal for the Semantic Web.

The *trust* problem is intimately related with the openness and distributed model that the Web promotes consequently results in communication and collaboration with strangers. Unlike client-server systems, where certain nodes can easily be distinguished as *trustworthy* under certain conditions, a decentralized-distributed network of nodes may provide no such guarantee. Moreover, several authors claim that "*trust and security are two sides of the same coin*" [8] and, the authors of [9] believe research in decentralized trust and reputation management is still in its infancy. According to [10] the next steps should focus on the adoption of a *broad notion of policy* that not only considers access control, but also looks at the effective combination of *policy rules* and *reputation models*, the inclusion of automated trust negotiation services for improving system interoperability, the introduction of a *controlled natural language*

[2] http://www.w3.org/2004/Talks/0412-RDF-functions/slide4-0.html

syntax for policy rules that helps users in the elaboration of policies, and the implementation of *explanatory* mechanisms for better understanding of the execution of trust systems.

The resolution of the set of issues listed above and the inclusion of a coordination model for application integration can contribute to the evolution of the Semantic Web into a *Semantic Enterprise Service Bus* (SESB). **Conceptual Spaces (CSpaces)** [11; 12] aims to contribute to this goal by suggesting a flexible conceptual and architecture model that can accommodate promising (and sometimes disperse) research initiatives, and outline general guidelines that can be followed by the Semantic Web research community. Section 2 provides an overview of what CSpaces is. In Section 3, a use case for integrating heterogeneous project management applications in the construction industry is presented. Related work and conclusions are presented in Section 4 and 5.

2 CSpaces Overview

Just as the Web has been characterized by an abstract model called REST (Representational State Transfer) [3] that is defined as a set of constraints (*client-server architecture, stateless, cache, uniform interface, layered system,* and *code-on demand*), CSpaces suggests to characterize the Semantic Web around seven building blocks [11]: *semantic data model, organizational model, coordination model, consensus-making model, security and trust model, knowledge access model,* and *architecture model.* In particular, CSpaces suggest the following approaches for the problems identified in the previous section of this document.

Dichotomy Problem. CSpaces promote to follow a *Semantic centric* approach instead of a *Web centric* approach in which machine processable semantics should be used more to model and not only to annotate data[3]. An intensive use of knowledge graphical visualization, controlled natural language[4], and natural language generation[5] techniques are the means proposed for facilitating information access.

Scalability Problem. The CSpaces paradigm proposes to partition the Semantic Web into a network of knowledge containers, and the creation of a reasoning space for each knowledge container in which approximation reasoning techniques [13] can be applied. Knowledge containers will be stored in superpeer [14] networks in which servers can reduce their workload by delegating on certain clients the execution of concrete tasks.

Heterogeneity Problem. Similar to [6, 7], local knowledge containers are connected by mapping and transformation rules and organized around influential knowledge containers that provide a shared view of the information stored in each of the local knowledge containers. This proposal avoids the problems of point-to-point mappings and merging approaches.

[3] Annotations represent a temporal solution from a *syntactic* Web towards a "*Semantic Centric*" Web. Annotations make a gradual and feasible transition between the two.

[4] Subset of a natural language (for instance English) with a domain specific vocabulary and a restricted grammar in the form of a small set of construction and interpretation rules. http://www.ics.mq.edu.au/~rolfs/controlled-natural-languages/

[5] http://www.aaai.org/AITopics/html/nlunder.html

Publishing Problem. CSpaces recommends the use of graphical visualization tools, controlled natural languages and natural language generation technology for facilitating the creation of machine processable semantics.

Inconsistency Problem. Each knowledge container maintains a reasoning space in which paraconsistent [15] reasoning and debugging techniques [16] can be applied.

Security and Trust Problem. CSpaces promotes a decentralized trust model in which policy rules and reputation trust models [9] are stored in each knowledge container. A representation of policy rules using controlled natural language [10] will make it easier for users to encode them. Thus, knowledge containers will create a network of spaces of trust.

Architecture Problem. CSpaces is not against the layer cake, but it does not commit also to it until a proper requirements analysis determines which formalisms are required by Semantic Web applications. On the other hand, CSpaces provide a more elaborated and refined specification for the Semantic Web than has been proposed till now.

Given space limitations, only five of the seven building blocks will be described in more detail in this section: semantic data model, organizational model, coordination model, security and trust model, and architecture model. A through description of the remaining building blocks can be found in [12]

2.1 Semantic Data Model

A **Conceptual Space** (CSpace) [11] is a knowledge container defined as a set of tuples[6]. In CSpaces each tuple has a well-defined structured that is represented by seven fields

<guid, fm, type, subspace, sguid, vguid, mguid>

Ideally, **fm** is a first order logical formula. However, limitations imposed by applications and/or members of the CSpace can restrict **fm** to less expressive formalism (like description logics, horn logic formulas and even RDF triples). The field **type** identifies in which formal language has **fm** been defined (e.g. fol, shiq_dl, horn_fol, dlp, etc). Unlike the Semantic Web, the current proposal of the CSpaces' semantic data model does not commit to a specific formal language until an exhaustive evaluation of use cases determines which languages are most appropriate for CSpaces (and by extension for the Semantic Web). The field **subspace** defines a subset of the CSpace in which each tuple belongs. Currently, there are seven different subspaces defined for each CSpace: domain theory (*dth*), metadata (*mtd*), instance (*itc*), trust and security (*tas*), mapping and transformation rules (*mtr*), annotations (*ant*), and subscriptions/advertisements (*sba*). The field **guid** is a global unique id[7] for the logical formula (which can simplify reification, and make the code more compact). The **sguid** field is the global unique identifier of the CSpace where they were created

[6] The CSpaces‘s semantic data model is strongly influenced by the CSpaces's coordination model. The coordination model is based on tuplespace computing paradigm that uses tuples as data container.

[7] Following W3C recommendations, http://www.w3.org/Addressing/Activity.html, the unique ids required in CSpaces will follow the URI/IRI specification.

(which attaches provenance to a logical formula). The field **vguid** is a version global unique identifier that distinguishes each version of a logical formula. The field (**mguid**) is the identifier of the member of the CSpace that stores the tuple. Given that each member of a CSpace has a reputation score, **mguid** can help to measure the degree of trustworthiness of each of the logical statements that are stored in a CSpace.

As it was mentioned before, a CSpace is subdivided into seven different subspaces. Each of these sub-spaces can have a "*mirror*" that stores an efficient representation (in terms of reasoning performances) of the data stored. Thus, each sub-space has a *raw* and *reasoning* side. All editing operations are done in the **raw** side and periodically the modifications are transferred to the **reasoning** side. The strict separation between raw and reasoning side allows the implementation of, for instance, *approximate reasoning* techniques [13] like language weakening and knowledge compilation for reducing the scalability problem and debug methods for eliminating *inconsistencies* [16]**.**

The sub-space **domain theory** stores a set of consistent logical theories which gives an explicit, partial account of a conceptualization. The sub-space **metadata** provides an ontological description of the CSpace itself. The sub-space **instances** is used to represent elements or individuals of concepts and the values of their attributes in a domain theory. The sub-space **annotations** define links between concepts and instances (topics) specified in each domain theory with information resources (occurrences). The sub-space **security and trust information** is described in terms of **policy rules** and **reputation** information [9]. The sub-space **mapping and transformation rules** define correspondences between common terms, relations and instances of two domain theories stored in an Individual and a Shared CSpaces. Finally, the sub-space **subscriptions and advertisements** stores queries that identify the information that is requested by information consumers and will be published by information producers. A more detailed description of those sub-spaces can be found in [12].

2.2 Organizational Model

Both [6, 7] have influenced the organizational model proposed by CSpaces. Both works promote the use of shared domain theories as an interlingua for data and application integration instead of using point-to-point translators. A second source of inspiration for the CSpaces's organizational model is the intuition that generation and sharing of machine processable semantics will follow a bottom-up approach (from personal knowledge specifications to shared knowledge specifications). The experiences reported in the EDAMOK project (http://edamok.itc.it/) following a distributed knowledge management approach [17] are reinforcing this intuition. Finally, the necessity to improve trustworthiness of the information stored motivates the creation of spaces of trust for a restricted group of agents (human users and applications).

Two types of CSpaces have been defined: *Individual* and *Shared* CSpaces [11]. The former is a knowledge container defined by an individual that reflects his/her own perception of a concrete domain. Shared CSpaces are conceptual spaces shared by several users that have reached an agreement on how to specify common domain theories, instances, annotations, etc. Shared CSpaces act as semantic bridges between several Shared and Individual CSpaces. A Shared CSpace can appear in three different flavors: *materialized* view, *virtual* view [18] and *hybrid materialized-virtual* view [19].

CSpaces are linked by mapping and transformation rules that currently are described using Distributed First Order Logic (D-FOL) [20]. D-FOL is a very flexible and rich formalism that is able to generalize and unify a variety of alignment frameworks like OIS [21] and e-connections [22]. Ideally networks of CSpaces should follow an organizational model similar to the one proposed in **CO4** (*Collaborative construction of consensual knowledge bases*) [23]. Thus, the configuration of a network of CSpaces will be defined by a DAG (*Directed Acyclic Graph*) model. This organizational model is very appropriate for building a network of distributed and related knowledge containers, because cyclic references are avoided and message flooding between nodes is reduced (critical for distributed queries). Also, a DAG configuration of CSpaces organized around Shared CSpaces match very well with a bottom-up approach for the generation and sharing of machine processable semantics. Given that the access to CSpaces is restricted to a set of users and a decentralized trust mechanisms will be implemented, Shared CSpaces will become on spaces of trust that will improve user confidence on the data stored by CSpaces networks. This is an important requirement for B2B, B2C and in general application integration.

2.3 Coordination Model

Like Triple-based computing [1], CSpaces also adopts the tuplespace computing paradigm [2] as the basics for its coordination model. Tuplespace computing is a very flexible and simple coordination model that matches very well with the philosophy of the Web of creating a global persistent space for publication of data. As opposite to the original tuplespace computing proposal in which tuples are not restricted to a concrete configuration (they are just ordered sets of heterogeneous objects), tuples in CSpaces follows the specification defined by the CSpaces's semantic data model. In addition, CSpaces promotes the use of rich and formal query languages for querying a CSpace instead of using only template matching. The query language is not fixed yet and it will depend of the representation language chosen to describe information in a CSpace. In addition, CSpaces' coordination model allow agents to write information in terms of their logical theories stored in their own individual CSpaces and not use the logical theory stored in the destination CSpaces. Thus, the coordination model is aware of this situation to request query rewriting and data transformation services. The TSpace API[8] has been adapted for reading (`take`, `waitToTake`, `read`, `waitToRead` and `scan`) and publishing (`write`) tuples in CSpaces. A detailed description of this API can be founded in [12].

The inability of tuplespace computing to provide *flow decoupling* from the client side [5] is solved by extending the tuplespace computing model with subscription operations [11]. Thus, two main roles for participants are defined: **producers**, which publish information and advertisements (description of which information will be published); and **consumers**, which expresses its interest in concrete information by publishing subscriptions. The new extensions based on SIENA API[9] can be founded in [12].

[8] http://www.almaden.ibm.com/cs/TSpaces/Version3/ClientProgrGuide.html

[9] http://www-serl.cs.colorado.edu/serl/siena/

Finally, transaction support is included to guarantee the successful execution of a group of operations (or the abortion of all of them if one fails). Transactions have been proposed in several tuplespace computing implementations like TSpaces and JavaSpaces[10]. The new extensions for transaction support can be founded in [12].

2.4 Security and Trust Model

The *security-trust model* proposed for CSpaces relies on three relevant works in the area of distributed trust: PACE (http://www.isr.uci.edu/projects/pace/), PROTUNE (http://rewerse.net/) and POBLANO (http://www.jxta.org/). PACE influenced the architecture style of the CSpaces' Trust Model. PROTUNE provides the first serious attempt of combining policy-based and reputation-based trust management approaches. Finally, POBLANO includes a refined implementation of reputation-based trust management described in [8]. Inspired partially in the PACE architecture, the architecture style of the CSpaces' Trust Model is defined around four services. The **Key Manager,** like in PACE**,** provides the necessary infrastructure for the generation of unique digital key pairs. The **Trust Manager**, following PACE and PROTUNE proposals, incorporates policy-based and reputation-based trust management capabilities. The **Execution Manager** schedules the execution of operations related to trust requests and monitors the trust manager service to avoid its collapse in case of many concurrent requests. Finally, the **Storage Manager** provides persistence for storing trust information. Each CSpace provides a subspace for maintaining trust-security data component and is responsible for maintaining the locally cached identity information stored in the Information layer. It may request public keys from other peers when needed and also respond to key revocation notifications.

These four services are hosted, similar to PACE, by peers following a decentralized and distributed approach, but in the case of CSpaces only heavy-clients and servers [11] are able to provide those services. Unlike PACE, the security-trust model is not a horizontal layer over the information and communication layer. The CSpaces's Security and Trust Model is a vertical layer that provides support to the CSpaces's Architecture Model, Semantic Data Model, Organizational Model, Coordination, and Consensus-making Model. For instance, encryption features are supported by the Key Manager at the architecture level, the Storage Manager is tightly related with the Semantic Data Model, and the Execution Manager supports the coordination model.

2.5 Architecture Model (Blue-Storm)

The preliminary proposal for CSpaces architecture, called *Blue-Storm* and briefly outlined in this section, is strongly influenced by the work done in OceanStore[11], Edutella[12] and SWAP[13]. In particular, the architecture combines pure P2P and client/server systems in a hybrid proposal called **super-peer** systems [14]. Like OceanStore, this configuration drives into two-tiered system. The upper-tier is composed of well-connected, powerful and always available group of servers, and the lower-tier, in

[10] http://java.sun.com/developer/products/jini/index.jsp

[11] http://oceanstore.cs.berkeley.edu/

[12] http://edutella.jxta.org/

[13] http://swap.semanticweb.org/public/index.htm

contrast, consists of clients (desktop computers, laptops, PDAs, mobile phones, etc) with limited computational resources that are only temporarily available. Three kinds of nodes are identified in CSpaces architecture: CSpace-servers, CSpace-heavy-clients and CSpace-light-clients.

CSpace-servers are responsible of storing primary and secondary replicas of the data published in Individual and Shared CSpaces; supporting services that keep track of the modifications to the content and explicitly identify versions of each (or part of each) Individual and Shared CSpaces stored in the server; providing an access point to the super-peer network and computational resources to execute read operations and publish information for CSpace-light-clients; maintaining and executing reasoning services for evaluating complex queries; running the coordination mechanism based on publish-subscribe and tuplespace computing; providing security and trust services; and monitoring and balancing workload between servers and clients.

CSpace-heavy-clients, on the other hand, are responsible for providing most of the services available in a CSpace-server (except workload balance and access point services), and also including presentation tools (based on Controlled Natural Language, Natural Language Generation and Knowledge visualization techniques) to facilitate the visualization and edition of knowledge contents. It is recommended that online clients have the obligation to share computational resources (CPU time, memory and persistent storage services). Thus CSpace-servers can divert client's resources demanding requests, and consequently, alleviate temporarily the workload of servers.

Finally, **CSpace-light-clients** represent only light-weight devices with limited computational resources like PDAs and mobile phones that only will include presentation tools to edit and visualize knowledge content stored on CSpaces. Given the limited computational resources of light-clients, the CSpaces that are accessed and edited by light-clients are hosted by CSpaces-servers. Thus, CSpaces-servers will allocate computational resources for CSpaces-light-clients.

3 Integration of Project Management Applications

The use case presented in this section has been slightly adapted from a previous work conducted by members of the Engineering Informatics Group (EIG)[14] of Stanford University and published in [24; 25].

The aim of the use case is to integrate heterogeneous applications for project management which teams of the construction industry use for dealing with large construction projects. The members of those teams can belong to different organizations and use diverse tools for the same purposes or for managing separate aspects of the project. Thus, large volumes of project information will be created from different sources geographically dispersed. In particular, Chen et al., aimed to integrate the following tools: Primavera Project Planner^{TM} (P3)[15] and Microsoft Project^{TM}[16] for scheduling, Vite SimVision^{TM}[17] for project organization, 4D Viewer [26] for the view of construction progress, and weather information from YAHOO (http://weather.yahoo.com/).

[14] http://eil.stanford.edu/

[15] http://www.primavera.com/

[16] http://office.microsoft.com/project/

[17] Vite was adquired by ePM: http://www.epm.cc/solutions/epmSV.htm

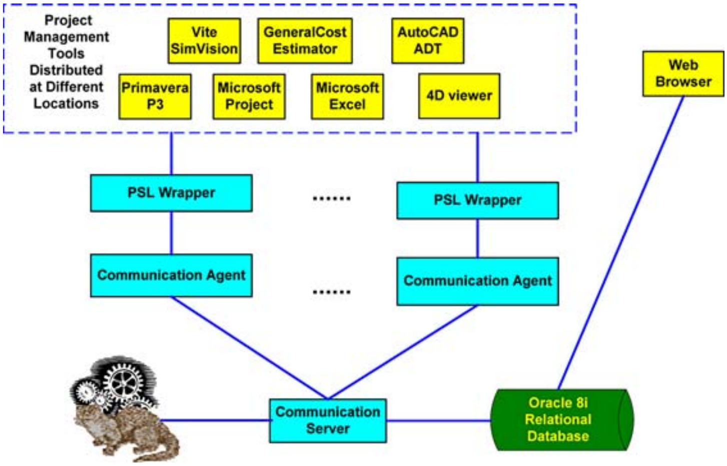


Fig. 1. Distributed architecture for application integration [25]

Following a pure materialized approach (*data warehouse*), [24; 25] use the PSL Ontology [7] for application integration. The PSL Ontology addresses the problem of integrating multiple process related applications by providing a standard language for process specification that offers a set of explicit and unambiguous definitions. In particular, this interlingua is specifically tailored for manufacturing systems. The underlying language used for the PSL Ontology is KIF (Knowledge Interchange Format) [27], a first order based language created for the exchange of knowledge among disparate computer programs.

To achieve interoperability between project management applications using the PSL Ontology, [24; 25] determine that several extensions of PSL are required, and it is also necessary to specify mapping and transformation rules between those applications and PSL.

[25] outlines a distributed architecture based on Java socket communication with two main components (Figure 1): a set of PSL wrappers for retrieving and transferring information from management applications to PSL and vice versa; and a back-end system that includes communication, storage and reasoning services. The theorem-prover Otter[18] is used for consistency checking and for scheduling constraint verification. For instance, [25] reports the experiment of changing the duration of activity ID120 (“Lay Foundation”) of the Arnold’s House project[19], if the Yahoo weather forecast service announces 5 days of rain. Otter was able to reschedule the tasks affected and provide a new consistent project schedule proposal.

3.1 Using CSpaces for Integrating Project Management Applications

The solution for application integration proposed by [24; 25] has been revised to test the suitability of CSpaces in these kind of scenarios. The work, currently in progress,

[18] http://www-unix.mcs.anl.gov/AR/otter/

[19] Arnold’s House project is one of the tutorial examples of Vite SimVision™

defines two Shared CSpaces and five Individual CSpaces. Each Individual CSpaces stores the information of each of the applications that will be integrated. The first Shared CSpace will store the original PSL Ontology, and the second will store the extended version of the PSL Ontology together with the data of each of the management applications. Individual CSpaces will be stored in desktop computers (heavy-clients) and the shared CSpaces will be hosted by a server. Each computer will run Otter 3.3 (provides the reasoning space), Oracle 10g (providing persistent storage services for CSpaces) and an extended version of ActiveSpace[20] (providing coordination services). Heavy-clients will also store PSL wrappers for extracting and transferring information from project management applications (Figure 2).

ActiveSpace follows a JavaSpace-like abstraction for building SEDA [28] style applications. SEDA (*Staged Event Driven Architecture*) is an architectural pattern for building massively scalable, distributed and concurrent systems. ActiveSpace substitute the communication server and the communication agents described in [25].

Several aspects of CSpaces have been simplified or not considered in order to facilitate a prototypical implementation:

- The domain theories stored on the server and the heavy-clients are essentially the same (PSL plus extensions for project management applications). Only some information related with internal activities (not relevant for the construction project), roles and costs are hidden and are not shared in the Shared CSpace. Thus, mapping and transformation rules are not required in this scenario
- The Yahoo Weather service used in [25] has been substituted by the National Weather service[21]. This experimental service offers also the possibility to retrieve 5-day forecast information using SOAP messages.
- Only the weather information is constrained by policy rules in order to guarantee that National Weather service will be the only agent that can change weather information
- Original PSL wrappers and versions of the project management applications will be used to avoid re-implementation of those modules
- Current Semantic Web standards like OWL and RDF will not be considered in this example. Data will be represented using FOL as the original use case proposed
- Only a materialized view approach for Shared CSpaces is considered
- Project Management applications are the only front-end tools considered. Graphical visualization tools and Natural Language Generation services are not included
- No light-clients are considered

The modifications that users introduce in project management applications are parsed into PSL formulas using PSL wrappers. The adapter component (Figure 2) is responsible to pack each PSL formula into a CSpace's tuple, and execute the

[20] http://docs.codehaus.org/display/AS/Home

[21] http://www.nws.noaa.gov/xml/. A WSDL specification of the interface of this service can be found at http://www.weather.gov/forecasts/xml/DWMLgen/wsdl/ndfdXML.wsdl

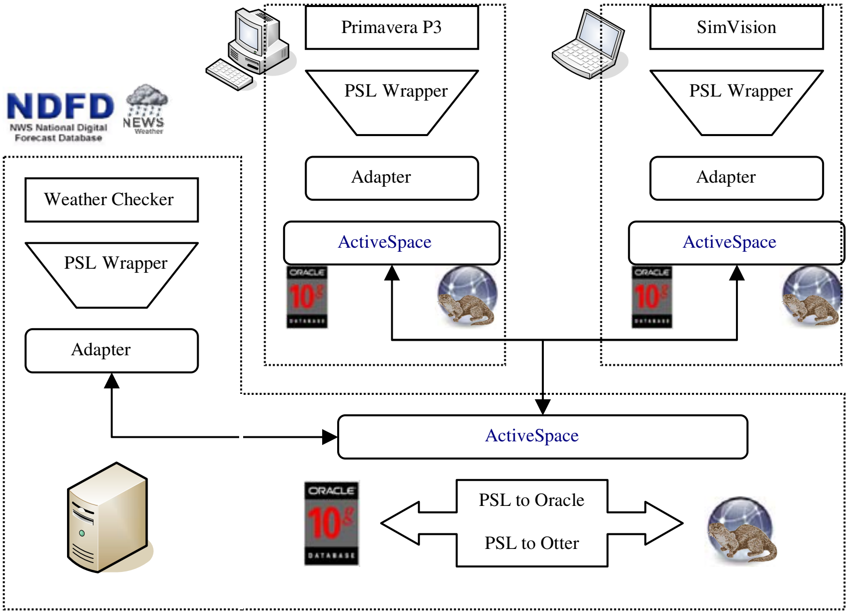


Fig. 2. Revised distribute integration infrastructure

appropriate write operation. The values of the fields guid, vguid and mguid of each tuple, are included by the coordination infrastructure (revised version of ActiveSpace). Each running instance of ActiveSpace(one for each heavy-client or server) is responsible for replicating new information to other ActiveSpace instances. For the user/application perspective, ActiveSpace generate a virtual shared memory in which applications can read and write information using a small collection of simple operations.

Each instance of ActiveSpace maintains a register of the operations that have been performed. Before the information is replicate, the operation registers of the CSpaces involved in the replication process are analyzed to determine if there are conflicts. In the presence of conflicts, replication is not performed until conflicts are solved by the users of each CSpace.

Persistent storage services required by each instance of ActiveSpace are provided by Oracle 10g that maintain a "raw" version of every CSpace. Like in [25], periodically the information stored in Oracle is transferred to a local Otter reasoning service that create the associated reasoning space in which inference techniques can be applied.

4 Related Work

Middleware is the "*glue*" that facilitates and manages the interaction between applications across heterogeneous computing platforms. *Remote Procedure Call* (RPC), *Transaction Processing* (TP) *Monitor*, *Message-Oriented Middleware* (MOM), *Message Brokers* and *Web Services* [29] have been proposed for creating middleware infrastructures. As a key component of Service Oriented Architecture (SOA), *Enterprise Service Bus* (ESB [30]) is a distributed infrastructure and is contrasted with solutions, such as broker technologies, which are commonly described as hub-and-spoke. ESB aims to provide in one infrastructure the three major styles of Enterprise Integration [30]: Service-oriented, Message-driven and Event-driven architectures. However, ESB is positioned as an infrastructure component, and as such as a component that does not host or execute business logic. This is in contrast to components such as service requesters, service providers and the Business Service Choreography whose role is to handle business logic. Common ESB capabilities are listed below [30]:

- Mediation or transformation of service messages and interactions
- Routing, Addressing, Publish / subscribe, Fire & forget, events and Synchronous and asynchronous messaging
- Authentication, Authorization, Non-repudiation, Confidentiality and end-to-end security.
- Transactions (atomic transactions, compensation, WS-Transaction)

CSpaces infrastructure can allocate most of the capabilities that ESB requires and in addition exploits the theoretical advantages of machine processable semantics. Thus, CSpaces can transform the Semantic Web into a *Web-scale Semantic Enterprise Service Bus* (SESB) in which *Semantic enabled Service Oriented Architectures* (SESOA) can be deployed.

On the other hand, the extension of tuplespace computing with machine processable semantics have been proposed by several parallel initiatives like Semantic Web Spaces [31] and sTuples [32]. The former has been proposed by the Freie Universität Berlin, and it was originally envisaged as a framework for modeling tuplespace-based communication for the Semantic Web stack. Only the coordination model is currently well-designed following a pure tuplespace computing specification in which notifications, publish-subscribe extensions and transaction support are not considered. sTuples [32], on the other hand, has been developed as part of the Pervasive Computing work at the Nokia Research Center. sTuples was built as an extension of Sun's JavaSpaces to share DAML+OIL instances in tuple fields for the purposes of supporting the semantic interoperability of heterogeneous and dynamic clients in a pervasive computing environment. Transactional support and publish-subscribe extensions have been added to the original coordination model implemented by JavaSpaces.

5 Conclusions

Like Triple Space Computing [1], **Conceptual Spaces (*CSpaces*)** [11; 12] aim to transform the Semantic Web into a global persistent information space for application integration and coordination. CSpaces improve Triple Space Computing by combining *tuplespace* and *publish-subscribe* paradigms, eliminating the *stateless requirement* of the REST abstract model, and promoting a new conceptual and architectural model for the *Semantic Web* organized around seven building blocks: *semantic data model, organizational model, coordination model, consensus-making model, security and trust model, knowledge access model,* and *architecture model.*

The use case suggested by [24; 25] for the integration of project management applications and adapted in Section 3 shows that CSpaces is a conceptual and architecture model flexible enough to accommodate many scenarios that have been proposed in the literature about data integration and application integration. The materialized view approach based on PSL extensions proposed by [24; 25] has been integrated in a network of CSpaces. On the other hand, CSpaces' coordination model simplifies the integration of applications by providing a simple but powerful coordination mechanism that reduces the implementation efforts required in [24; 25]. CSpaces also include a security and trust model in which policy rules can be specified for restricting access to certain information. In addition, the **super-peer** architecture provides a decentralized and distributed infrastructure in which heavy-clients can alleviate the workload of servers and allow users to work offline without losing the advantages of machine processable semantics support.

Acknowledgements

I would like to thank Uwe Keller and Mick Kerrigan (DERI Innsbruck, Austria), Stefan Decker, Armin Haller and Eyal Oren (DERI Galway), Manfred Hauswirth (Ecole Polytechnique Federal de Lausanne, EPFL), Kincho Law, Michael Genesereth, Charles Petri (Stanford University) and the members of the Stanford Logic Group for fruitful discussion and feedback. The work is partly funded by the European Commission under the NoE KnowledgeWeb (IST-2004- 507482)

References

1. D. Fensel: Triple Space computing: Semantic Web Services based on persistent publication of information, the IFIP Int'l Conf. on Intelligence in Communication Systems, 2004.
2. Gelernter, D.: Generative Communication in Linda. ACM Transactions on Programming Languages and Systems, 7(1):80–112, 1985
3. Fielding, R. T.: Architectural styles and the design of network-based software architectures. PhD Thesis, University of California, Irvine, 2000
4. Khare, R., and Taylor, R. N.: Extending the Representational State Transfer (REST) Architectural Style for Decentralized Systems. Proceedings of the International Conference on Software Engineering (ICSE), May, 2004, Edinburgh, Scotland.
5. Eugster, P. T., Felber, P.A., Guerraoui, R., Kermarrec, A. M.: The Many Faces of Publish/Subscribe. ACM Computing Survey (2003).
6. de Bruijn, J., Martín-Recuerda, F., Manov D. and Ehrig, M.: State-of-the-art survey on Ontology Merging and Aligning V1. Project Deliverable d4.2.1, 2004. SEKT project IST-2003-506826 (http://sekt.semanticweb.org/)
7. Schlenoff, C., Gruninger, M., Tissot, F., Valois, Lubell, J., Lee, J.: The Process Specification Language (PSL): Overview and Version 1.0 Specification. NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD., 2000.
8. A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of* 33rd Hawaii International Conference on System Sciences, 2000.

9. Suryanarayana, G., and Taylor, R.: A Survey of Trust Management and Resource Discovery Technologies in Peer-to-Peer Applications. ISR Technical Report UCI-ISR-04-6, July 2004. http://www.isr.uci.edu/tech_reports/UCI-ISR-04-6.pdf
10. Bonatti, P. A., et al. Semantic web policies - a discussion of requirements and research issues. In 3rd European Semantic Web Conference (ESWC), Lecture Notes in Computer Science, Budva, Montenegro, June 2006. Springer.
11. Martín-Recuerda, F.: Towards CSpaces: A new perspective for the Semantic Web. In Proceedings of the 1st International IFIP/WG12.5 Working Conference on Industrial Applications of Semantic Web (IASW 2005). Jyvaskyla, Finland. August, 2005
12. Martín-Recuerda, F., Nixon, L. J. B., Bontas, E. P.: D2.4.8.1: Technical and ontological infrastructure for Triple Space Computing v1. Knowledge Web report. December 2005. http://knowledgeweb.semanticweb.org/
13. Wache, H., et al. Scalability state of the art of ontology based technology. Knowledge Web report. December 2004. http://knowledgeweb.semanticweb.org/
14. Yang, B., and Garcia-Molina, H.: Designing a Super-peer Network. IEEE International Conference on Data Engineering, 2003
15. J. Beziau, What is paraconsistent logic, in: Batens D., Mortensen C.,Priest G. and Van Bendegem J.P. (eds). Frontiers of paraconsistent logic. Research Studies Press: Baldock, 2000, 95-111.
16. S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In Proceedings of the eighteenth International Join Conference on Artificial Intelligence, IJCAI'03. Morgan Kaufmann, 2003.
17. M. Bonifacio, P. Bouquet and R. Cuel. "Knowledge Nodes: the Building Blocks of a Distributed Approach to Knowledge Management". Journal of Universal Computer Science, 8(6), 652-661. 2002
18. Ullman, J. D.: Information Integration Using Logical Views. ICDT 1997: 19-40
19. Hull, R., Zhou, G.: A Framework for Supporting Data Integration Using the Materialized and Virtual Approaches, In Proc. ACM SIGMOD '96, Montreal, Canada, 1996.
20. Ghidini C., and Serafíni, L. Distributed first order logic - revised semantics. Technical report, ITC-irst, January 2005.
21. Calvanese, D., De Giacomo, G., and Lenzerini, M.: A framework for ontology integration. In Isabel Cruz, Stefan Decker, Jerome Euzenat, and Deborah McGuinness, editors, The Emerging Semantic Web, pages 201–214. IOS Press, 2002.
22. Cuenca Grau, B., Parsia, B., and Sirin, E.: Working with multiple ontologies on the semantic web. In Proceedings of the Third Internatonal SemanticWeb Conference (ISWC2004), volume 3298 of Lecture Notes in Computer Science, 2004.
23. Euzenat, J.: Building consensual knowledge bases: context and architecture, in N. Mars (ed.), Towards very large knowledge bases, IOS press, Amsterdam (NL), pp143-155, 1995
24. Cheng, J., and Law, K. H.: Using Process Specification Language for Project Information Exchange. Proceedings of the 3rd International Conference on Concurrent Engineering in Construction, Berkeley, CA, pp. 63-74, 2002.
25. Cheng, J., Gruninger, M., Sriram, R. D., and Law, K. H.: Process Specification Language for Project Information Exchange. International Journal of Information Technology in Architecture, Engineering and Construction, 1(4):307-328, 2003.
26. McKinney, K. and Fischer, M.: Generating, Evaluating and Visualizing Construction Schedules with 4D-CAD Tools. Automation in Construction, Vol. 7, No. 6, pp. 433-447, 1998
27. Genesereth, M.R., and Fikes R.: Knowledge Interchange Format 3.0. Technical Report KSL-92-01, Knowledge Systems Laboratory, Stanford University. 1992

28. Welsh, M., Culler, D., and Brewer, E.: SEDA: An Architecture for Well-Conditioned, Scalable Internet Services. In Proceedings of the Eighteenth Symposium on Operating Systems Principles (SOSP-18), Banff, Canada, October, 2001
29. Alonso, G., Casati, F., Kuno, H., and Machiraju, V.: Web Services. Springer, 2004.
30. Chappell, D.: Enterprise Service Bus. O'Reilly Media, Inc. June 2004.
31. Tolksdorf, R., Paslaru-Bontas, E., and Nixon, L. J. B.: Towards a tuplespace-based middleware for the Semantic Web. IEEE/WIC/ACM International Conference on Web Intelligence WI2005, Compiegne University of Technology, France, September 2005
32. Khushraj, D., Lassila, O., and Finin, T.: sTuples: Semantic Tuple Spaces. In Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous'04)

A New Evaluation Method for Ontology Alignment Measures

Babak Bagheri Hariri and Hassan Abolhassani

Institute for Studies in Theoretical Physics and Mathematics (IPM), and
Semantic Web Research Laboratory, Computer Engineering Department,
Sharif University Of Technology, Tehran, Iran
hariri@ce.sharif.edu, abolhassani@sharif.edu

Abstract. Various methods using different measures have been proposed for ontology alignment. Therefore, it is necessary to evaluate the effectiveness of such measures to select better ones for more quality alignment. Current approaches for comparing these measures, are highly dependent on alignment frameworks, which may cause unreal results. In this paper, we propose a framework independent evaluation method, and discuss results of applying it to famous existing string measures.

1 Introduction

Ontology alignment is an essential tool in semantic web to overcome heterogeneity of data, which is an integral attribute of web. In [1] heterogeneities are divided to syntactic, terminological, conceptual and semiotic/pragmatic. However, most of the existing alignment methods are focused on discovering terminological and conceptual heterogeneities between ontologies. Various measures for discovering similarities between pairs of entities has been proposed. It is necessary to compare them to gain better alignment methods by selecting better ones.

We divide current measures in three major groups. First group, referred to as *Terminological Matching*, contains string measures which try to find the similarity of entities according to the texts used in them. These techniques are based on the fact that the same concepts are likely to be modeled using quite similar names. In the current paper we will evaluate the *Levenshtein* distance [2] which used the *Edit Distance* to match two strings, the *Needleman-Wunsch* distance[3], which assigns a different cost on the edit operations, the *Smith-Waterman* [4], which additionally uses an alphabet mapping to costs, the *Monge-Elkan* [5], which uses variable costs depending on the substring gaps between the words , the *Stoilos* similarity [6] which try to modify existing approaches for entities of an ontology, *Jaro-Winkler* similarity [7,8], which counts the common characters between two strings even if they are misplaced by a "short" distance, and the *Sub-string* distance [9]which searches for the largest common substring. Other two groups are *Synonymity* similarity which tries to find the similarities based on using dictionary or linguistic ontologies like *Wordnet*, and *Hierarchical* similarities which try to find the similarity of the entities considering their state in the ontology graph.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 249–255, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Today there are some evaluation methods for ontology alignment. Most of them try to compare ontology alignment techniques to find a suitable one. In [10, 11, 12, 13] most of these methods are described. In these methods after defining some measures like *Precision*, *Recall*, performance and memory, results of alignment frameworks are compared. However, there are limited works that try to evaluate the measures used in frameworks and not the frameworks themselves. In [6] it is tried to compare string measures, and select best one. The main problem of this approach, however, is its dependency to the underlying alignment framework it uses. The distinguishing feature of our approach is to have an evaluation method independent of the alignment framework. The results of applying it to the string measures (first group) on EON_{2004} [14] training set is discussed in the paper.

The rest of the paper is organized as follows. In Sect. 2 the proposed approach, named *Direct Evaluation Technique* is introduced. Sect. 3 shows the results of applying it to compare string measures and Sect. 4 is a discussion and conclusion.

2 Direct Evaluation Technique

Fig. 1 displays the main characteristic of our approach compared to the existing ones. Existing approaches make evaluations on the results of alignment; hence they have dependency to the alignment method. This means that the quality of their comparison is directly related to the way different measures are implemented in them. On the other hand, our approach evaluates measures directly by using a technique named *Sensitivity Analysis* [15] which is used in Data Mining discipline. Therefore we named our approach *Direct Evaluation Technique.*

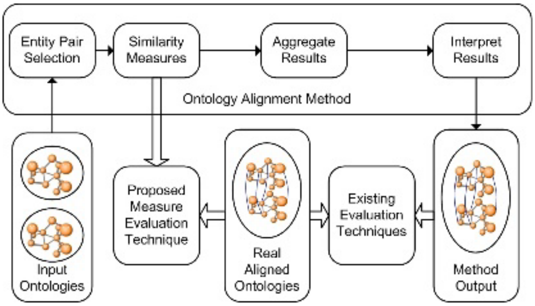


Fig. 1. Proposed evaluation technique versus existing techniques

2.1 Using Neural Networks

Neural Network sensitivity analysis allows us to measure the relative influence that each attribute has on the output result. The *Sensitivity Analysis* proceeds as follows [15]:

1. Generate a new observation x_{mean}, with each attribute value in x_{mean} equal to the mean of the various attribute values for all records in the test set.

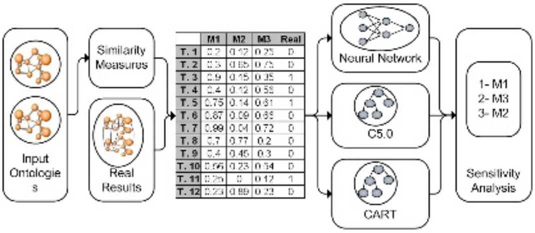
2. Find the network output for input x_{mean}. Call it $output_{mean}$.
3. Attribute by attribute, vary x_{mean} to reflect the attribute minimum and maximum.
4. Find the network output for each variation and compare it to $output_{mean}$.

The Sensitivity Analysis will find that varying certain attributes from their minimum to their maximum will have a greater effect on the resulting network output than it has for other attributes.

To apply Sensitivity Analysis to the measure evaluation problem, we create a matrix in which rows represent relation between an entity from the first ontology to an entity of the second one. Columns show the similarity value for two entities as given by corresponding similarity measure, and the last column is the actual similarity value (0 or 1). In this formulation the last column represents the target variable and other columns are predictors. Now the problem is reduced to a data mining problem in which we are interested to know the effect of each predictor on the target variable and therefore it is possible to apply Sensitivity Analysis for this problem. Fig. 2 shows the process.

2.2 Other Sensitivity Analysis Techniques

As discussed in [15] best evaluation may not be selected by a single judge alone. Instead, one should seek a confluence of results from a suite of different models. Other techniques which is used for sensitivity analysis are *CART*[1] and $C_{5.0}$ decision trees [15]. They are using different mathematical basis. CART bases its decisions on the *goodness of split* criterion, that $C_{5.0}$ applies an *information-theoretic approach*, and that Neural Network bases their learning on *back-propagation*. Yet these three different algorithms represent streams that broadly speaking, have come together, and forming a confluence of results. In this way, the models act as validation for each other.



	M1	M2	M3	Real
T. 1	0.2	0.12	0.25	0
T. 2	0.3	0.65	0.75	0
T. 3	0.9	0.15	0.35	1
T. 4	0.4	0.12	0.52	0
T. 5	0.75	0.14	0.61	1
T. 6	0.87	0.09	0.65	0
T. 7	0.99	0.04	0.72	0
T. 8	0.7	0.77	0.2	0
T. 9	0.4	0.45	0.3	0
T. 10	0.56	0.23	0.34	0
T. 11	0.25	0	0.12	1
T. 12	0.23	0.89	0.22	0

Fig. 2. Proposed evaluation technique in detail

We use combination of these three methods to rank the measures used in alignment techniques. We evaluate the measures separately in each of the three methods, and interpret the outputs by finding measures which all of them agree upon. The whole process is shown in Fig. 2.

[1] Classification and regression trees.

Table 1. Related importance of measures using Neural Networks

Test	Lev.	Need.	Smit.	Jaro.	Mong.	Stol.	Sub.
103	0.296	0.166	0.141	0.004	0.003	0.037	0.215
204	0.203	0.148	0.232	0.093	0.068	0.084	0.183
205	0.458	0.021	0.126	0.082	0.052	0.023	0.040
221	0.304	0.167	0.160	0.008	0.006	0.021	0.215
222	0.274	0.156	0.076	0.010	0.007	0.048	0.172
223	0.206	0.003	0.032	0.026	0.004	0.027	0.098
224	0.288	0.157	0.148	0.009	0.014	0.037	0.277
302	0.588	0.310	0.074	0.102	0.089	0.047	0.056
303	0.262	0.028	0.295	0.036	0.033	0.035	0.151
304	0.337	0.119	0.023	0.008	0.012	0.059	0.169
All	0.361	0.180	0.250	0.121	0.013	0.023	0.045

Lev.=Levensteine, Need.=Needleman-Wunsch, Smit.=Smith-Waterman, Jaro=Jaro-Winkler, Mong.=Mong-Elkan, Stol.=Stolios, Sub.=SubString

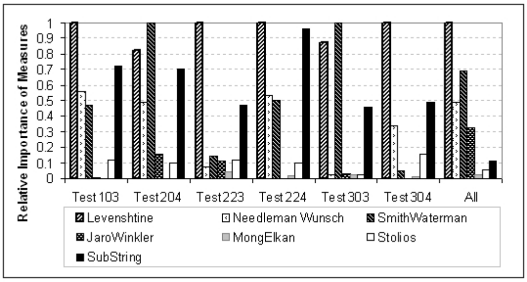


Fig. 3. Evaluation of string measures using Neural Networks

3 Results

We have examined our proposed evaluation on major string similarity methods using EON_{2004} [14] test set. The evaluation sets we used are following:

- **Group 1_{xx}:** We only use test 103 from this group. This test compares the ontology with its generalization in OWL Lite. Names of entities in this group is remaining without any changing and cause this group not to be a suitable data set for evaluation of string measures.
- **Group 2_{xx}:** The reference ontology is compared with a modified one. Tests 204, 205, 221, 223 and 224 are used from this group. Modifications involved naming conversions like replacing the labels with their synonyms as well as modifications in the hierarchy. We use these tests as a training set.
- **Group 3_{xx}:** The reference ontology is compared with four real-life ontologies for bibliographic references found on the web and left unchanged. We use tests 302, 303 and 304 from this group. This is the only group which contains real tests and may be the best one for evaluation of measures.

Table 2. Most 4 important measures

Neural Network	CART	C5.0
Levenshtein	Jaro-Winkler	Needleman-Wunsch
SubString	Levensteine	Levensteine
Smith-Waterman	Monge-Elkan	Jaro-Winkler
Needleman-Smith	Stoilos	Monge-Elkan

- **All:** To have a larger test set, we merged all the data from described data sets. The table of results for each data set, which is introduced in Sect. 2, are concatenated to each other and form a larger data set.

Each data set contains some entities. Name of each entity is compared with the names of all other entities. Each comparison of two strings is assigned a similarity degree. Every entry for a string contains a key which is purposed for the identification of the correctness of a pair. After collecting output for each measure, we evaluate them for each data set as it is described in Sect. 2.

3.1 Evaluating String Measures Using Proposed Framework

Table 1 displays results of applying similarity analysis on each test set using *Clementine* [2] tool. In this table each row shows the *relative importance* of measures used in the corresponding data set. As it is clear from the table, *Levenshtein* similarity is the most important one in predicting the relation of entities. Fig. 3 shows the relative importance of measures in each data set, by normalizing the results of Table 1. We have other evaluations based on decision trees. In Table 2 we compare results of these techniques. All of three tests agree about importance of Levenshtein similarity on the test set. Neural Network chooses *Levenshtein* while $C_{5.0}$ and *CART* select it as second suitable measure. While this experiment shows that *Levenshtein* similarity has the best behavior across described measures, it may not be the most suitable measure for other domains. It is necessary to evaluate measure on different domains to find their real behavior in different situations.

3.2 Indirect Evaluation of String Measures

We have implemented a simple ontology alignment framework to evaluate previously mentioned measures in a real alignment framework. *Jena* [3] is used for parsing input ontologies. In this implementation we haven't used any Structural or Hierarchical measures and evaluation is only based on previously mentioned string measures. In the interpretation phase we make the assumption that each entity of the first ontology is correspondent to at most one of the entities of the other ontology. In each test we use one of the measures and calculate the *Harmonic Mean* of *Precision* and *Recall* of each test as *F-Measure* [10].

Fig. 4 shows the results of applying each measure on data sets 302, 303 and 304. As it is clear one can't priorities any of the measures using results of this test.

[2] http://www.spss.com/clementine

[3] http://jena.sourceforge.net/

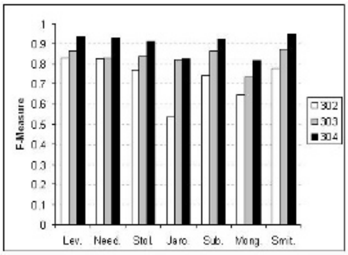


Fig. 4. Indirect evaluation by implementing measures in a framework

In [6], using such a framework on the same data sets, has different results, which confirms our claim that evaluation of a framework may not necessarily shows the effectiveness of the measures used in it. Results of evaluation of measures used in a framework even if the framework is as simple as possible, may be influenced by other parts of the underlying framework itself. Therefore two frameworks may not choose a same measure as the best one on a same data set. On the other hand direct evaluation of string measures as described in Sect. 2 has not such problems and can be used to find best measures in each domain and results in better frameworks for ontology alignment.

4 Conclusion and Future Works

In this paper the main problem of existing evaluation frameworks, which we refer to as indirect evaluation, for ontology alignment has been discussed and it is claimed that the results of such evaluations are influenced by underlying framework itself. To overcome this problem, a new evaluation framework which is based on data mining techniques is proposed, featuring independence of alignment framework. It is discussed how an evaluation problem is reduced to a data mining problem of *Sensitivity Analysis*. As an example, evaluation of famous string measures on a data set is demonstrated. To have more accurate results, we should study behavior of ontology alignment measures on different domains and various data sets using the introduced method.

Acknowledgements

This research was in part supported by a grant from I.P.M. (No. CS1384-4-04).

References

1. Bouquet, P., Ehrig, M., et all: Specification of a Common Framework for Characterizing Alignment. Technical Report deliverable 2.2.1, Knowledge Web, (Statistical Research Division, Room 3000-4, Bureau of the Census, Washington, DC, 20233-9100 USA)
2. Levenshtein, V.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics-Doklady **10** (1966) 707–710

3. Needleman, S., Wunsch, C.: A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of two Proteins. Molecular Biology **48** (1970) 443–458
4. Smith, T., Waterman, M.: Identification of Common Molecular Subsequences. Molecular Biology **147** (1981) 195–197
5. Monge, A.E., Elkan, C.P.: The Field-Matching Problem: Algorithm and Applications. In: Proceedings of the second international Conference on Knowledge Discovery and Data Mining. (1996)
6. Stoilos, G., Stamou, G., et all: A String Metric for Ontology Alignment. In: Proceedings of the ninth IEEE International Symposium on Wearable Computers. (2005) 624–237
7. Jaro, M.: Probabilistic Linkage of Large Public Health Data Files. Molecular Biology **14** (1995) 491–498
8. Winkler, W.E.: The State Record Linkage and Current Research Problems. Technical Report RR99/04, U. S. Bureau of the Census, Statistical Research Division, Room 3000-4, Bureau of the Census, Washington, DC, 20233-9100 USA, (1999)
9. Euzenat, J., Bach, T.L., et all: State of the Art on Ontology Alignment. Technical Report deliverable 2.2.3, Knowledge Web, (Statistical Research Division, Room 3000-4, Bureau of the Census, Washington, DC, 20233-9100 USA)
10. Euzenat, J., Ehrig, M., et all: Benchmarking Methodology for Alignment Techniques. Technical Report deliverable 2.2.2, Knowledge Web, (Statistical Research Division, Room 3000-4, Bureau of the Census, Washington, DC, 20233-9100 USA)
11. Do, H., Melnik, S., et all: Comparison of Schema Matching Evaluations. In: Proceedings of the 2nd Int. Workshop on Web Databases, (German Informatics Society)
12. Euzenat, J.: Towards Composing and Benchmarking Ontology Alignments. In: Proceedings of the ISWC-2003 workshop on semantic information integration. (2003) 165–166
13. Rahm, E., Bernstein, P.A.: A Survey of Approaches to Automatic Schema Matching. The Very Large Databases Journal **10**(4) (2001) 334–350
14. Sure, Y., Corcho, O., et all, eds.: Proceedings of the 3rd Evaluation of Ontology-based tools (EON). (2004)
15. Larose, D.T.: Discovering Knowledge In Data. John Wiley and Sons, New Jersey, USA (2005)

Representing and Reasoning with Application Profiles Based on OWL and OWL/XDD

Photchanan Ratanajaipan[1], Ekawit Nantajeewarawat[2], and Vilas Wuwongse[3]

[1] Computer Science, School of Technology, Shinawatra University, Pathumthani, Thailand
photchanan@shinawatra.ac.th
[2] Sirindhorn Intl. Inst. of Technology, Thammasat University, Pathumthani, Thailand
ekawit@siit.tu.ac.th
[3] Computer Science and Information Management,
Asian Inst. of Tech., Pathumthani, Thailand
vw@cs.ait.ac.th

Abstract. An application profile specifies a set of terms, drawn from one or more standard namespaces, for annotation of data, and constrains their usage and interpretations in a particular local application. A framework for representing and reasoning with application profiles using the OWL and OWL/XDD languages is proposed. The former is a standard Web ontology language and the latter is a definite-clause-style rule language that employs XML expressions as its underlying data structure. Constraints are defined in terms of rules, which are represented as XDD clauses. Application of the approach to defining an application profile with fine-grained semantic constraints is illustrated. A prototype library metadata validation system has been implemented.

1 Introduction

With the rapid growth of information resources, the importance of metadata as a key tool for resource description and management is increasingly recognized. For its interoperability, e.g. applicability in a context other than its origins and common agreement on its usage, metadata itself needs to be described using standards or schemas. An application profile provides a schema of metadata, i.e., a vocabulary for explaining data in an application domain. It specifies a set of terms, drawn from one or more standard namespaces (schemas), and constrains possible interpretations of them. These terms are used for describing various properties of resources, and are often refined, extended, or constrained to meet the requirements of a local application.

Resource Description Framework (RDF) Schemas and XML Schemas have been used in [10] and [8], respectively, for defining application profiles. Their capability to describe restrictions on properties are however rather limited; moreover, neither of them provides a mechanism for describing implicit relationship between domain elements, which is desirable for specifying semantic constraints of several kinds. Inspired by the basic idea presented in [14], this paper presents a new approach to representation of application profiles based on the OWL Web ontology language [11] and the XML Declarative Description (XDD) theory [13]. OWL can be seen as an extension of RDFS schemas; it provides a rich set of concept constructors, which

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 256 – 262, 2006.
© Springer-Verlag Berlin Heidelberg 2006

readily lend themselves to expressing many kinds of property-restriction refinement. As a complement to the strength of OWL in describing the structure of a domain and structural constraints, the XDD theory provides extensive rule-oriented facilities for describing implicit information and reasoning with domain elements. Among other things, XDD allows one to describe fine-grained semantic constraints on relationships between property values, e.g. "DateIssued of a library resource may not be prior to DateCreated of it", and to define derivable properties, e.g. "Every Creator of a resource that is a part of a collection is a Contributor of that collection". Such semantic constraints and implicit properties cannot be represented using OWL alone.

To illustrate application of the proposed approach, a formalization of an application profile based on Dublin Core Metadata Initiative's library application profile (DC-Lib) using OWL/XDD is presented. The possibility of extending the profile by describing semantic constraints and implicit relationships between properties is demonstrated. To begin with, Section 2 reviews the concept of application profiles. After briefly recalling OWL, Section 3 introduces XDD and OWL/XDD. Section 4 shows application of the proposed framework in the context of library metadata validation system, along with some experimental results.

2 Application Profiles

An application profile defines terms for annotation of data with metadata, and describes how to use the terms consistently with specific rules of usage. A profile serves as a schema—more precisely, a metadata schema—which describes a set of metadata and types of values. Heery and Patel [7] have defined an application profile as "a schema that consists of data elements drawn from one or more namespaces, combined together by implementers, and optimized for a particular local application".

Duval [5] elaborates shared principles and practicalities of metadata, which outline the basic requirements for application profiles. Profiles should be able to combine data elements from multiple namespaces by supporting declaration of them. Such declaration assures that terms have unique definitions within the bounds of a namespace. Moreover, profiles should facilitate declaration of the following constraints and restrictions.

- *Obligation constraints:* They provide cardinality restrictions of elements, specifying, for example, whether the elements are optional, mandatory, or conditional. The data element identifier, for instance, may be specified to be mandatory.
- *Data schemes and values constraints:* Constraints of this kind provide a mechanism for refinement of a standard. They are useful when a standard is too loose in regard to the values required for a data element, or when further restrictions on the value space is needed in the context of an application. For example, the value space of the element coverage describing a resource may be restricted to a specific set of some named places that are relevant to that resource in a particular domain.
- *Relationship and dependency specification:* Specification of interrelationships between data elements and their values should also be supported. For instance, the presence of one element may require the presence of another element. A dependency constraint may also restrict the value space of one element based on the value

of another element. For example, if the value of the element type of a resource is "audio", then the value of the element format of this resource cannot be "HTML".

As an example, DC-Library Application Profile (DC-Lib), proposed by DCMI-Library Working Group [4], has been widely employed for clarifying the use of the Dublin Core Metadata Element Set in the domain of libraries and library-related applications. Metadata elements in DC-Lib are taken from DCMI Metadata Terms [3] and MODS [9], e.g. Title, Alternative (title), Creator, Contributor, Publisher, Date, DateCopyrighted, and DateSubmitted. Refinements are made to these elements, for example, by imposing on them some simple existence constraints such as mandatory (M), mandatory if applicable (MA), optional (O) and required (R), along with constraints on allowed number of occurrences. Encoding schemes and values for some elements are also defined. DC-Lib is presented in a tabular style conformed to Dublin Core Application Profile Guidelines [6]. In the literature, application profiles have been represented using a variety of formalisms with varying degrees of formality and precision, ranging from natural languages, tabular formats, to RDF and XML Schemas. The degree of interoperability of their metadata increases as their underlying formalism has more precise and machine-processable semantics. Owing to their formal semantics and XML-based syntax, OWL and XDD are promising languages for representation of application profiles.

3 OWL and OWL/XDD

OWL (Web Ontology Language) [11] is a machine-processable knowledge representation language designed for describing ontologies (schemas and their instances). OWL provides expressive class constructors, which can be used for describing a sufficient condition and/or a necessary and sufficient condition for membership of a class based on restriction on the values of properties that individuals belonging to a class can take. As such, it is suitable for describing the structure of an application domain and structural constraints thereon.

However, the capability of OWL with respect to describing relationships between individuals is rather limited due to the restricted ability of its underlying logic. This area is a stronghold of rules, which offer extensive facilities for representing and reasoning with individuals and relations between them. Such facilities are required for specifying semantic constraints of many kinds. This shortcoming is overcome in the proposed approach by employment of extensive rule-oriented facilities provided by the XML Declarative Description (XDD) theory [12,13]. Of central importance to this theory, it uses XML expressions as their underlying data structure; consequently, one can seamlessly specify information to be extracted as well as implicit information to be derived from OWL elements and other forms of XML data.

An *XDD description* is a set of *XDD clauses*, each of which is a formula of the form $H \leftarrow B_1, \ldots, B_m, \beta_1, \ldots, \beta_n$, where $m, n \geq 0$, H and the B_i are XML-expressions, and the β_j are constraints. H is called the *head*, and $\{B_1, \ldots, B_m, \beta_1, \ldots, \beta_n\}$ the body of the clause. XML *expressions* are XML elements that are extended by incorporation of variables. Variables of several kinds, with different syntactical usage and

```
<ap:appProfile rdf:about=" http://shinawatra.ac.th/libprofile">
  <dc:title>SIU Library Application Profile</dc:title>
  <ap:uses><owl:DatatypeProperty rdf:about="&dc;title"/></ap:uses>
  <ap:uses><owl:DatatypeProperty rdf:about="&dc;created"/></ap:uses>
   ...
   <owl:DatatypeProperty rdf:about="&dc;title">
      <rdfs:label>Title</rdfs:label>
      <dc:type rdf:resource="&dcp;element"/>
      <ap:obligation>M</ap:obligation>
  </owl:DatatypeProperty>
   <owl:DatatypeProperty rdf:about="&dc;created">
      <rdfs:label>Created</rdfs:label>
      <dc:type rdf:resource="&dcp;element-refinement"/>
      <ap:obligation>MA</ap:obligation>
      <ap:hasEncodingScheme>ISO8601</ap:hasEncodingScheme>
  </owl:DatatypeProperty>
  ...
</ap:appProfile>
```

Fig. 1. OWL-Based Representation of a Part of SIU Library Application Profile

different instantiation characteristics, are employed. It is assumed that a name-variable (*N*-variable), a string-variable (*S*-variable), an attribute-value-pair-variable (*P*-variable), and an XML-expression-variable (*E*-variable) are prefixed with $N:, $S:, $P: and $E:, respectively. A *constraint* is an expression that specifies certain restriction on XML elements or their components. The reader is referred to [12,13] for the formal semantics of XDD descriptions.

OWL/XDD [14] is a language that combines OWL and XDD, allowing OWL elements to contain variables and their relationships to be expressed as constraints and rules. It incorporates OWL into XDD by basing XDD's constructs on OWL/XML elements. Concrete examples of OWL/XDD clauses will be given in the next section.

4 Examples

In this section, an example of application profile, based on Dublin Core Metadata Initiative's library application profile (DC-Lib), represented using OWL will be given. Constraints, restrictions and rules defined using OWL/XDD clauses are illustrated.

The OWL-based representation in Fig.1 provides a description of the application profile *http://shinawatra.ac.th/libprofile*, called "SIU Library Application Profile". It employs annotation elements defined in existing standard namespaces such as the RDF Schema (prefixed by rdfs:), DCMI Metadata Terms (prefixed by dc:), DCMI Grammatical Principles (prefixed by dcp:) etc., and uses elements from a user-defined namespace, DC-Lib profile schema (prefixed by ap:). The profile uses terms and encoding schemes defined in some other namespaces. Terms such as title, created, location are encoded using owl:DatatypeProperty elements. The profile describes the term created, for instance, by asserting that its type is "element refinement", its encoding scheme is "ISO8601", and its obligation constraint is "MA" ("mandatory if applicable").

A constraint in a profile can be expressed as an OWL/XDD clause. The OWL/XDD clause in Fig. 2, for example, defines the meaning of an obligation constraint "M", i.e., if a property is mandatory, then the minimum cardinality restriction

```
C1: <owl:Class rdf:ID="$S:ClassX">
       <rdfs:subClassOf>
          <owl:Restriction>
             <owl:minCardinality rdf:datatype="&xsd;#int">1
             </owl:minCardinality>
             <owl:onProperty>
                <owl:DatatypeProperty rdf:resource="$S:PropertyP"/>
             </owl:onProperty>
          </owl:Restriction>
       </rdfs:subClassOf>
    </owl:Class>
       ←  <owl:DatatypeProperty rdf:about="$S:PropertyP">
             $E:PropertyP1
             <ap:obligation>M</ap:obligation>
             $E:PropertyP2
          </owl:DatatypeProperty>,
          <owl:DatatypeProperty rdf:about="$S:PropertyP">
             $E:PropertyP3
             <rdfs:domain rdf:resource="$S:ClassX "/>
             $E:PropertyP4
          </owl:DatatypeProperty>,
          <dl:ConceptSubsumes subsumer="rdfs:Resource" subsumee="$N:ClassA"/>
```

Fig. 2. An XDD Clause Representing a Constraint

```
C2: <xdd:DateConflict instance="$S:X" created="$S:DateC"
     issued="$S:DateI"/>
       ←  <$N:ClassA rdf:about="$S:X">
             $E:PropertyP1
             <dc:issued>$S:DateI</dc:issued>
             $E:PropertyP2
          </$N:ClassA>,
          <$N:ClassA rdf:about="$S:X">
             $E:PropertyP3
            <dc:created>$S:DateC</dc:created>
             $E:PropertyP4
          </$N:ClassA>,
          <dl:ConceptSubsumes subsumer="rdfs:Resource"
           subsumee="$N:ClassA"/>
          <DatePrior date1="$S:DateI" date2="$S:DateC"/>.
```

Fig. 3. An XDD Clause Representing a Constraint

is 1. Data schemes/values constraints and relationship/dependency specification can also be described by OWL/XDD clauses. The clause C_2 in Fig. 3, for example, specifies the constraint that DateIssued cannot be prior to DateCreated. Furthermore, implicit elements can be defined through OWL/XDD clauses. For example, the clause C_3 in Fig. 4 asserts that every creator of a resource that is a part of a collection is a contributor of the collection.

Based on the proposed framework, a prototype library metadata validation system has been implemented. Computation with OWL/XDD is carried out using the XET engine [15]—an interpreter system that supports computation by equivalent transformation [1] in the context of XDD—and a Description Logic reasoner for OWL, called RACER.

Fig. 5 shows a part of an RDF library metadata used for testing the prototype system. The clauses in Fig. 2, 3 and 4 are used for validating this library metadata. From the clause C_1 in Fig. 2, the XET engine derives the cardinality constraint repre-

```
C3: <rdfs:Resource rdf:about="$S:A">
        <dc:contributor>$S:CreatorC</dc:contributor>
   </rdfs:Resource>
        ←  <rdfs:Resource rdf:about="$S:B">
               $E:PropertyP1
               <dc:creator>$S:CreatorC</dc:creator>
               $E:PropertyP2
           </rdfs:Resource>,
           <rdfs:Resource rdf:about="$S:B">
               $E:PropertyP3
                <dc:isPartOf rdf:resource=$S:A/>
               $E:PropertyP4
           </rdfs:Resource>.
```

Fig. 4. An XDD Clause for Deriving Implicit Property

```
<rdfs:Resource rdf:about="dclib:resource1">
   <dc:title>Ancient City</dc:title>
   <dc:creator>Giordano, Frank C.</dc:creator>
   <dc:isPartOf rdf:resource ="dclib:resource1"/>
</rdfs:Resource>
<rdfs:Resource rdf:about="dclib:resource2">
   <dc:creator>Martin, James C.</dc:creator>
</rdfs:Resource>
<owl:Class rdf:ID="dclib:Book">
   <rdfs:subClassOf rdf:resource="rdfs:Resource"/>
</owl:Class>
<dclib:Book rdf:about="dclib:book1">
   <dc:title>The work of giants : rebuilding Cambodia</dc:title>
   <dc:creator>Wenk, Brian., Rain, Nick.</dc:creator>
   <dc:created>2005-12-15</dc:created>
   <dc:issued>2006-01-30</dc:issued>
</dclib:Book>
```

Fig. 5. Metadata of Library Resources

```
<owl:Class rdf:ID="http://www.w3.org/2000/01/rdf-schema#Resource">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality rdf:datatype="&xsd;#int">1</owl:minCardinality>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:resource="&dc;#title"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Fig. 6. An Element Derived Using the Clause C_1 in Fig. 2

sented as the OWL element in Fig. 6. Using this obtained constraint, OWL reasoning tools can detect that the rdfs:Resource identified by dclib:resource2 has no title, which causes metadata inconsistency. The clause C_2 in Fig. 3 specifies a constraint on the properties DateCreated and DateIssued for the class rdfs:Resource and all of its subclasses. The rdfs:subClassOf axiom allows an OWL engine to derive the information that dclib:Book is one such subclass. Accordingly, a conflict between DateCreated and DateIssued in the resource dclib:book1 can be detected. Note that from the clause C_3 in Fig. 4, it is derivable that a contributor of dclib:resource1 is "Martin, James". As a result, no inconsistency is reported for this resource, even when the obligation constraint on contributor is "mandatory" and no explicit contributor is defined for dclib:resource1.

5 Conclusion

The paper proposes a framework for representation of and reasoning with application profiles based on the OWL and OWL/XDD language. The expressive power of those languages allows one to define terms and several kinds of constraints, including semantic constraints on interrelationships between data elements and values. Further works include refinement of the framework for describing more complicated metadata structure, and more thorough evaluation of the performance of the prototype system in a wider variety of application domains.

References

1. Akama, K. and Nantajeewarawat, E.: Formalization of the Equivalent Transformation Computation Models. Journal of Advanced Computational Intelligence and Intelligent Informatics 10 (2006) 245-259
2. Baker, T., Dekkers, M., Heery, R., Patel, M., Salokhe, G.: What Terms Does Your Metadata Use? Application Profiles as Machine-Understandable Narratives. Journal of Digital Information 2(2) (2001)
3. DCMI Metadata Terms. Available: http://dublincore.org/ documents/dcmi-terms/
4. DCMI-Libraries Working Group. (2004, September). Library Application Profile (2004-09-10). Available: http://dublincore.org/ documents/2004/09/10/library-application-profile
5. Duval, E.: Metadata Principles and Practicalities. D-Lib Magazine 8 (4) (2002). Available: http://www.dlib.org/dlib/april02/04contents.html
6. Dublin Core Application Profile Guidelines. Available: ftp://ftp.cenorm.be/PUBLIC /CWAs/e-Europe/MMI-DC/cwa14855-00-2003-Nov.pdf
7. Heery, R., Patel, M.: Application Profiles: mixing and matching metadata schemas. Ariadne 25(2000). Available: http://www.ariadne.ac.uk/issue25/app-profiles
8. Hunter, J.: An XML Schema Approach to Application Profiles (2000). Available: http://archive.dstc.edu/maenad/ appln_profiles.html
9. Metadata Object Description Schema (MODS). Available: http://www.loc.gov/mods
10. SCHEMAS Registry. Available: http://www.schemas-forum.org/registry
11. Smith, M. K., Welty, C., McGuinness, D. L.: OWL web ontology language guide, W3C Recommendation, 10 February 2004. http://www.w3.org/TR/owl-guide
12. Wuwongse, V., Akama, K., Anutariya, C., Nantajeewarawat, E.: A Data Model for XML Databases. Journal of Intelligent Information Systems 20 (2003) 63–80
13. Wuwongse, V., Anutariya, C., Akama, K., Nantajeewarawat, E.: XML Declarative Description: A Language for the Semantic Web. IEEE Intelligent Systems (2001) 54–65
14. Wuwongse, V., Yoshikawa, M.: Towards A Language for Metadata Schemas for Interoperability. Proc. of International Conference on Dublin Core and Metadata Applications 2004 (DC2004), Shanghai, China, October 11-14, 2004, 21–25
15. XML Equivalent Transformation. Available: http://kr.cs.ait.ac.th/XET

OWL-Full Reasoning from an Object Oriented Perspective

Seiji Koide[1,2] and Hideaki Takeda[1]

[1] National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430
koide@grad.nii.ac.jp, takeda@nii.ac.jp
http://www.nii.ac.jp/
[2] Galaxy Express Corporation, 1-18-16, Hamamatsu-cho, Minato-ku, Tokyo 105-0013
koide@galaxy-express.co.jp
http://www.galaxy-express.co.jp/

Abstract. Bridging the gap between OWL and Object-Oriented Programming (OOP) languages is an indispensable condition to enable the Object-Oriented Modeling in Software Engineering by OWL. However it is very difficult in case of static OOP languages like Java and C#. We have developed SWCLOS, which is an OWL processor seamlessly built on top of Common Lisp Object System (CLOS), a dynamic OOP language. SWCLOS allows programmers to develop application domain models by OWL and enables OOP upon the models. In this paper, we explain the semantic gap between OWL and OOP languages, introduce the RDFS and OWL realization at SWCLOS, and discuss the OWL features from OOP perspectives. Finally we demonstrate the OWL-Full level performance in SWCLOS.

1 Introduction

It is natural to combine the domain modeling in Object-Oriented Programming (OOP) with the idea of *object-centered modeling* in ontology development. Recently, the Software Engineering Task Force (SETF) in W3C Semantic Web Best Practices and Deployment Working Group has started to promote synergies between the Semantic Web and the domains associated with Software Engineering[1]. One of the objectives is the Ontology Driven Software Engineering, in which ones expect benefits of unambiguous domain models, consistency checking facilities, validated model sharing, and semi-automatic code generation in software development. The realization of the Ontology Driven Architecture (ODA) by SETF requires to reorganize the object-oriented modeling for Software Engineering on the framework of Semantic Web, in particular, OWL. In order to enable the OOP upon OWL, we should bridge the semantic gap between OOP languages and OWL. However it is difficult in case of static OOP languages like Java and C#. Rather we need dynamic OOP languages.

The problem of OWL-DL is the separation between the class and the individual from the viewpoint of Software Engineering. In reality, the decision whether

[1] http://www.w3.org/2001/sw/BestPractices/SE/

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 263–277, 2006.
© Springer-Verlag Berlin Heidelberg 2006

we capture an entity in a model as class or individual depends upon the characteristics of the application domain and the attitudes of human modelers. For example, a wine product such as ElyseZinfandel should be an individual for wine expert systems, but should be a class in logistics for wine wholesalers. Borgida, et al. [Borgida2003] pointed out that one must create a "meta-individual" in order to work around such problems in Description Logic. Still, there are no dominant ideas to compute OWL-Full by means of Description Logics such as Tableau Algorithms.

We developed an OWL processor called SWCLOS[2] [Koide2004, Koide2005] that is built on top of Common Lisp Object System (CLOS), a dynamic OOP language of Lisp. In CLOS, the class is not only an object schema to define instances but also an object per se called *metaobject*. CLOS programmers can encode meta-modeling using the CLOS reflective programming facilities and the Meta-Object Protocol [Kiczales1991]. Therefore, the OWL-Full performance can be obtained by CLOS meta-programming facilities using SWCLOS. In fact, the property owl:sameAs, of which domain is owl:Thing, can be attached to OWL classes in SWCLOS, because OWL classes are individuals of owl:Thing in SWCLOS. Thus, the lisp predicate `owl-same-p` is applicable to not only OWL individuals but also OWL classes.

On the contrary, the loss of Tableau Algorithms from Description Logic inference brings the incompleteness to the subsumption calculation [Nardi2003]. We carefully implemented the *extended structural subsumption algorithm* in SWCLOS. However, the completeness is not obtained yet.

In this paper, at Section 2 we explain the problem of OOP languages with the comparison to OWL/RDF, and the dynamic features of CLOS language that enable RDFS/RDF semantics. At Section 3, we introduce OWL specific features in SWCLOS and the *extended structural subsumption algorithm.* At Section 4, we demonstrate OWL-Full meta-modeling in SWCLOS, then we conclude at Section 5.

2 A Comparison of OWL/RDF and Object-Oriented Programming Languages

The Software Engineering Task Force compared OWL/RDF features to ordinary Object-Oriented Programming Languages (OOPLs) such as Java and C# [SETF2006]. They pointed out serious discrepancies between OOPLs and OWL/RDF as follows. Note that the class in OOPLs is compared to the OWL class, and the instance is compared to the OWL individual. The property and value pair (or role and filler pair) in OWL/RDF is compared with the slot or the member variable of OOPLs.

- Classes in OOPLs are regarded not as sets to which instances belong but as types for instances.
- Each instance in OOPLs belongs one class as its type's instance.

[2] It is available from http://pegasus.agent.galaxy-express.co.jp/galexinfo/indexe.htm

- Instances in OOPLs cannot change their type at runtime.
- The list of classes in OOPLs must be fully known at compile-time and cannot change after that.
- There is no reasoner in OOPLs that can be used for classification and consistency checking at runtime or build-time.
- Properties in OOPLs are defined locally to a class and not stand-alone entities.
- Instances in OOPLs cannot have arbitrary values for any property without the definition in its class, and no domain constraint.

However, some of these items are not properly applied to CLOS. We summarized the dynamic features of CLOS in Object-Oriented Programming as follows.

- Multiple Class Inheritance: Methods and slots are inherited from multiple classes.
- Dynamic Programming: CLOS provides the means to redefine class definitions in program runtime.
- Meta-Object: A class is the first-class entity as object in CLOS, so a class in CLOS is called *metaobject*.
- Meta-Class: A meta-class or a class of classes allows ones to modify methods for classes including system intrinsic methods using the Meta-Object Protocol [Kiczales1991].
- Reflective Programming: The behavior of meta-classes including system methods is alterable using the Meta-Object Protocol. A programmer can modify behaviors of lisp systems. For example, so-called NEW method can be customized adapting for the features of applications by programmers.

We have implemented OWL/RDF semantics with CLOS by leveraging such dynamic and reflective language features. In the rest of this section, we explain the implementation of basic OWL/RDF semantics and RDFS/RDF axioms and entailments in SWCLOS through CLOS features. OWL specific semantics and the implementation are explained in the next section.

2.1 The Type in CLOS and the Membership in RDF

A class in OWL/RDF is a set of some individuals (called an *extension*), and the class-subclass relation in OWL is the inclusiveness of the extensions. Namely, the statement that a class C_2 includes a class C_1 ($C_1 \sqsubseteq C_2$) means that all individuals of class C_1 are concurrently individuals of class C_2. On the other hand, the semantics of class-instance in CLOS is different from OWL/RDF. A class in CLOS is a thing of which instances share methods and slot structure definitions. The semantics of CLOS class is built on the frame of slot structures and methods. However, the class-subclass relation and class-instance relation in CLOS work upon the transitivity and subsumption just same as RDFS. In practice, the RDF entailment rule **rdfs9**[3] (subsumption rule) and **rdfs11**[4] (transitivity rule

[3] http://www.w3.org/TR/rdf-mt/#rulerdfs9

[4] http://www.w3.org/TR/rdf-mt/#rulerdfs11

on rdfs:subClassOf) are natively realized in the CLOS class-subclass relation. Therefore, OWL individuals are straightforwardly mapped to CLOS instances and OWL classes are mapped to CLOS classes. Thus, rdfs:subClassOf is replaced with class-subclass relation in CLOS and rdf:type is replaced with class-instance relation.

2.2 Multiple Types by Invisible Classes

In the semantics of OWL/RDF, an instance can belong to multiple classes. For example, a vintage wine vin:SaucelitoCanyonZinfandel1998 in Wine Ontology[5] is an instance of both vin:Vintage and vin:Zinfandel. However, a CLOS class is a prototype to create its instances, then instances must inevitably belong to a single class. To solve this problem, we have introduced the invisible class that may be a subclass of visible multiple classes. For example, vin:SaucelitoCanyonZinfandel1998 is an instance of `vin:Zinfandel.15` that is invisible in OWL and a subclass of vin:Vintage and vin:Zinfandel in CLOS.

2.3 Forward Reference by Proactive Entailments

In order to enable forward-referencing, CLOS automatically creates an undefined but referred class as a class under `forward-referenced-class`. However, an attempt to make an instance of a forward referenced class causes an alarm in CLOS. The forward referenced class must be defined by the time of its instance creation. This function is insufficient for RDF forward reference. Fortunately, there are explicitly a number of RDF and RDFS entailment rules, in addition to the monotonicity principle in Semantic Web. Therefore, if we encounter an undefined class reference in reading an OWL file, we can create it as the most abstract concept in the context by applying various RDF and RDFS entailment rules for the context without the contradiction in definitions that will appear later on. For instance, **rdf1**[6] can be utilized for an undefined predicate to be created as an instance of rdf:Property, and **rdfs4**[7] assures for a subject and an object in triple to be defined as a resource object. The definition afterwards may be used to refine forward-referencing definitions precisely. The dynamic OOP features of CLOS such as class-change and reinitialization in runtime enable the implementation upon the forward reference by means of such *proactive entailments.*

2.4 The Realization of RDFS/RDF Axioms and Entailments

We implemented all RDFS/RDF axioms and entailment rules[8] in SWCLOS by exploiting the CLOS potential with Meta-Object Protocol. Most of entailments are realized only by mapping RDFS classes to CLOS classes and RDFS

[5] http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine.rdf

[6] http://www.w3.org/TR/rdf-mt/#rulerdf1

[7] http://www.w3.org/TR/rdf-mt/#rulerdfs4

[8] http://www.w3.org/TR/rdf-mt/

metaclasses (rdfs:Class and rdfs:Datatype) to CLOS metaclasses. In RDFS, rdfs:Class is an instance of itself. Such membership loop is the source of reflective systems and `cl:standard-class` in CLOS is also an instance of itself. However, CLOS does not allow to include other membership loops except `cl:standard-class`, so we worked around this problem by setting another class of rdfs:Class and making customized `typep` that pretends the membership loop upon rdfs:Class.

```
(cl:typep rdfs:Class rdfs:Class)    -> common-lisp:nil
(cl:subtypep (class-of rdfs:Class) rdfs:Resource) -> t
(typep rdfs:Class rdfs:Class)       -> t
```

Where `t` means boolean true in Lisp, and `typep` is a type testing function that is almost same as Common Lisp native predicate `cl:typep` except on `rdfs:Class`.

3 OWL Reasoning in SWCLOS

3.1 OWL Axioms over RDFS Axioms

In theory, OWL is an extension of RDFS/RDF. Therefore, SWCLOS syntactically and semantically reads the OWL definition file[9] as RDFS/RDF, and keeps RDFS/RDF semantics among OWL vocabularies in RDFS vocabularies. The followings demonstrate the relation that is defined in the OWL definition between rdfs:Class and owl:Class.

```
(typep owl:Class rdfs:Class)    -> t
(subtypep owl:Class rdfs:Class) -> t
```

In the CLOS perspective, owl:Class is also a metaclass as well as rdfs:Class, because it is a subclass of rdfs:Class. The class in CLOS defines slot structures or the role existence in its instances. Thus, the above axioms involve that owl:Class inherits the roles for rdfs:Class instances and rdfs:Resource instances (rdfs:Resource is a superclass of rdfs:Class). Namely, the roles such as rdfs:comment, rdfs:label, and rdfs:subClassOf can be attached to instances of owl:Class.

However, there exist some ambiguities to include OWL vocabularies among RDFS vocabularies. We set several axioms in addition to the defined ones in the OWL definition file. See Table 1 in Appendix. **Axiom1** is a compromise between OWL theory and the reality. In OWL-Full theory, owl:Thing is unified to rdfs:Resource, and then we cannot distinguish them. However, **axiom1** is needed in reality, just same as owl:Class is identified to a subclass of rdfs:Class in the OWL definition file. **Axiom2** is crucial for OWL-Full. The instances of owl:Class inherits the roles of owl:Thing (and rdfs:Resource). Thus, every class in OWL can have role owl:sameAs, owl:differentFrom, etc., as OWL individuals.

[9] http://www.w3.org/2002/07/owl.rdf

3.2 Anonymous Restriction Classes for Properties

The OWL object-centered expressions look like objects rather than RDF graphs, while they still obey RDF syntax and semantics. Therefore, the property restrictions in OWL/RDF turn out anonymous classes as instances of owl:Restriction. Then, the subjective CLOS object in the expression is defined as a subclass of the restriction classes that appears within rdfs:subClassOf or owl:intersectionOf representation forms.

In the CLOS perspective, a subclass inherits roles that exist in its superclasses, so it is reasonable that an anonymous restriction class, which provides the information of property value constraint, is placed at the superclass position of the subjective CLOS object. The information of restriction is inherited and shared by all instances of the subclasses. The slot information for instance such as value restriction (owl:allValuesFrom) for CLOS slots are defined in the direct class or its superclasses of an instance, and the information is stored into the CLOS *slot definition objects* that belong to the defined class.

The CLOS native type facet in the slot definition is utilized to realize the value restriction (owl:allValuesFrom) and the existential restriction (owl:someValuesFrom) in OWL. On the other hand, in order to implement cardinality restrictions for property value (owl:maxCardinality, owl:minCardinality, and owl:cardinality), we have introduced new slot facets, `mincardinality` and `maxcardinality`, into the *slot definitions.* If there exist multiple pieces of information upon a property with same restriction but different values among superclasses, they are collected and reduced to the most special one according to the monotonicity principle. For instance, the most special concepts are computed for the value restriction or the existential restriction and the maximum `mincardinality` and minimum `maxcardinality` are calculated for the cardinality restriction. When SWCLOS creates new instances, those constraints stored in the effective slot definition does work as constraints in instance creation. Thus, the satisfiability-checking for slot-value is performed in instance creation.

3.3 Axiomatic Complete Relations

Among many properties in OWL, only `owl:intersectionOf`, `owl:unionOf`, `owl:complementOf`, and `owl:oneOf` make axiomatic assertions. In other words, these properties define the complete equivalency upon the binary relation of concepts. For example, the following asserts the definition of `WhiteBordeaux` from the right-hand side to the left-hand side, and if something is a `Bordeaux` and `WhiteWine`, it is concluded to be a `WhiteBordeaux`.

$$\texttt{WhiteBordeaux} \equiv \texttt{Bordeaux} \sqcap \texttt{WhiteWine}$$

Similarly, the following assertion defines `WineColor`, which has the enumerative membership of `White`, `Rose`, and `Red`, so that the instance of `WineColor` is exactly one of the three, and not to be the others.

$$\texttt{WineColor} \equiv \{\texttt{White Rose Red}\}$$

Therefore, it is not necessary to mind the open world assumption upon such axiomatic complete relation properties. If we find the right-hand side of such equation matches the database, then we may conclude the left-hand side without worry about other statements.

See the following example. SWCLOS concludes that QueenElizabethII should be a woman, because it is asserted that a person who has gender female is a woman, and it is also asserted that QueenElizabethII is an instance of Person and hasGender Female. Here note that SWCLOS proactively made the entailment without demand or query from users.

```
(defIndividual Female (rdf:type Gender) (owl:differentFrom Male))
                                                 -> #<Gender Female>
(defResource Person (rdf:type owl:Class)
  (owl:intersectionOf
    Human
    (owl:Restriction (owl:onProperty hasGender)
                     (owl:cardinality 1))))      -> #<owl:Class Person>
(defResource Woman (rdf:type owl:Class)
  (owl:intersectionOf
    Person
    (owl:Restriction (owl:onProperty hasGender)
                     (owl:hasValue Female))))    -> #<owl:Class Woman>
(defIndividual QueenElizabethII (rdf:type Person)
  (hasGender Female))                       -> #<Woman QueenElizabethII>
```

3.4 Substantial Properties and Non-substantial Properties

There are many properties that rule the inclusiveness of concepts, i.e., rdfs:subClassOf, owl:intersectionOf, owl:unionOf, owl:equivalentClass, owl:equivalentProperty, etc. From the viewpoint of DL, they have same strength for subsumption decidability. However, from the viewpoint of Ontology Engineering and Software Engineering, we have to discriminate substantial ones and non-substantial ones for ruling subsumption. Borgida [Borgida2003] argued that ones should deal with individual objects that remain related rather than volatile references. Mizoguchi [Mizoguchi2004] has claimed that the IS-A relation (the substantial sorts) should comply with single inheritance from the viewpoint of Ontology Engineering, whereas an object may have multiple roles (the non-substantial sorts). Kaneiwa and Mizoguchi [Kaneiwa2005] developed the formal ontology on property classification and extended Order-Sorted Logic onto the property classification.

It is also important from the ontology and database maintainability to distinguish persistent relations and temporal relations. In SWCLOS, rdfs:subClassOf relation is mapped onto class-subclass relation, and a CLOS object as rdfs:subClassOf property value is additionally placed in the direct-superclasses-list slot of the class object. However, in case that a property p1 is a subproperty of or a equivalent property of rdfs:subClassOf, whether should we place the p1's value into the direct-superclasses slot in the class or not? In other words, what property in OWL should cause the structural variation in the CLOS class-subclass relation, and what property should cause subsumption reasoning without the structural variation? In SWCLOS, we specified that rdfs:subClass, owl:intersectionOf, and owl:unionOf should cause the variation, but owl:equivalentClass, owl:equivalentProperty and other properties, including subproperties and equivalent properties of rdfs:subClass, owl:intersectionOf, or owl:unionOf, should affect the inference but not the structural variation.

Conversely, we should define the substantial and persistent subsumption with rdfs:subClassOf, owl:intersectionOf, and owl:unionOf, and the non-substantial subsumption should be defined through other properties. The substantial subsumption may cause the proactive entailment, but the non-substantial subsumption should not cause any structural variation in the entailment. Thus, such discrimination of substantial and non-substantial subsumption allows us to add and delete relations and keeps it easy to maintain ontologies.

3.5 Extended Structural Subsumption Algorithm

The structural subsumption algorithm is described as follows [Baarder2003] for the $\mathcal{FL}_0$ level, which allows only conjunction ($C \sqcap D$) and value restriction ($\forall R.C$).

Let

$$A_1 \sqcap \ldots \sqcap A_m \sqcap \forall R_1.C_1 \sqcap \ldots \sqcap \forall R_n.C_n$$

be the normal form of the $\mathcal{FL}_0$-concept description C, and let

$$B_1 \sqcap \ldots \sqcap B_k \sqcap \forall S_1.D_1 \sqcap \ldots \sqcap \forall S_l.D_l$$

be the normal form of the $\mathcal{FL}_0$-concept description D, then $C \sqsubseteq D$ iff the following two conditions hold:
(1) For all i, $1 \leq i \leq k$, there exists j, $1 \leq j \leq m$ such that $A_j \sqsubseteq B_i$
(2) For all i, $1 \leq i \leq l$, there exists j, $1 \leq j \leq n$ such that $S_i = R_j$ and $C_j \sqsubseteq D_i$

The substantial inclusiveness is computed through the CLOS class-subclass relationship, and the non-substantial inclusiveness is deduced by the extended structural subsumption algorithm. The top concept $\top$ (owl:Thing) substantially subsumes every concept in the CLOS class-subclass relation, but the bottom concept $\bot$ (owl:Nothing) is virtually subsumed by other concepts through this extended structural subsumption algorithm. We extended the above structural subsumption algorithm in the $\mathcal{FL}_0$ level to what includes disjointness(owl:disjointWith), negation(owl:complementOf), equivalency(owl:sameAs, owl:equivalentClass, owl:equivalentProperty), functional and inverse-functional relation(owl:Functional-Property and owl:InverseFunctionalProperty), full existential-restriction(owl:some-ValuesFrom), filler restriction(owl:hasValue), and number restriction(owl:max-Cardinality, owl:minCardinality, and owl:cardinality) as follows.

1. If C is $\bot$, then $C \sqsubseteq D$ for any D, where $D \in$ owl:Class.
2. If D is $\top$, then $C \sqsubseteq D$ for any C, where $C \in$ owl:Class.
3. If D is $\bot$, then $\neg(C \sqsubseteq D)$ for any C, where $C \in$ owl:Class.
4. $\tilde{C}$ denotes a member of the equivalence group of C. If $\tilde{C} \sqsubseteq \tilde{D}$ (substantially), then $C \sqsubseteq D$ (inferred), where $\{C, D\} \in$ owl:Class.
5. $\ncong$ denotes complement relation in `complementOf`. If $\tilde{C} \ncong \tilde{D}$, then $\neg(C \sqsubseteq D)$.
6. Collect all substantially subsuming concepts and restrictions (all of CLOS superclasses) for each $\tilde{C}$ and each $\tilde{D}$, instead of C and D, and do the structural comparison as follows. Hereafter, use the notation such as $A_1 \sqcap \ldots \sqcap A_m \sqcap \forall R_1.C_1 \sqcap \ldots \sqcap \forall R_n.C_n$ for $\tilde{C}$ and $B_1 \sqcap \ldots \sqcap B_k \sqcap \forall S_1.D_1 \sqcap \ldots \sqcap \forall S_l.D_l$ for $\tilde{D}$, i.e. in case of value restrictions.
   $\bar{R}$ denotes a member of equivalent property group of R and $\approx$ denotes the equivalency in `equivalentProperty`. For all i, $1 \leq i \leq k$, if there exists j, $1 \leq j \leq m$ such that $A_j \sqsubseteq B_i$, and for all i, $1 \leq i \leq l$, there exists j, $1 \leq j \leq n$ such that $\bar{S}_i \approx \bar{R}_j$ and the followings hold, then $C \sqsubseteq D$.
   1) in case of value restriction ($\forall \bar{R}_j.C_j$ and $\forall \bar{S}_i.D_i$), $C_j \sqsubseteq D_i$.
   2) in case of full existential restrictions ($\exists \bar{R}_j.C_j$ and $\exists \bar{S}_i.D_i$), $C_j \sqsubseteq D_i$. This is incomplete but almost effective.
   3) in case of ($\forall \bar{R}_j.C_j$ and $\exists \bar{S}_i.D_i$), $C_j \sqsubseteq D_i$. This is incomplete but almost effective.
   4) in case of ($\exists \bar{R}_j.C_j$ and $\forall \bar{S}_i.D_i$), $C_j \sqsubseteq D_i$. This is incomplete but almost effective.
   5) in case of filler restrictions (owl:hasValue, $\bar{R}_j : a_j$ and $\bar{S}_i : b_i$), $a_j \sqsubseteq b_i$. Note that this is the inclusiveness among individuals. See Item 7.
   6) in case of ($\bar{R}_j : a_j$ and $\forall \bar{S}_i.D_i$), $a_j \in D_i$.
   7) in case of ($\bar{R}_j : a_j$ and $\exists \bar{S}_i.D_i$), $a_j \in D_i$. This overestimates the restriction but it is useful in most cases.
   8) in case of cardinality restriction ($\geq n\bar{R}_j$, $\leq nn\bar{R}_j$ and $\geq m\bar{S}_i$, $\leq mm\bar{S}_i$), $n \geq m$, $nn \leq mm$. This is incomplete with the combination of full existential restriction.
7. $\dot{C}$ denotes a member of `sameAs` group of C and $\doteq$ denotes the equivalency in `sameAs` relation. If $\dot{C} \doteq \dot{D}$, or $\dot{C}$ is transitive-lower than $\dot{D}$ on a shared transitive property, then $C \sqsubseteq D$, where $\{C, D\} \in$ owl:Thing. Note that the functional property entailment rule **rdfp1** and the inverse functional property entailment rule **rdf2** in [Horst2005] are used here.

Where Procedure 4 includes the performance of the subsumption test in RDFS semantics and the relation of owl:intersectionOf and owl:unionOf in OWL universe. Procedure 7 treats objects as individuals, including OWL classes. Obviously, this algorithm involves the recursion, but the calculation terminates. It is because CLOS prevents the terminological cycle in subsumption (ex. $C \sqsubseteq D$ and $D \sqsubseteq C$). While the occurrence cycle happens in chase of definitions with the combination of rdfs:subClassOf and owl:unionOf (ex. $B \sqsubseteq C$ and $C \equiv A \sqcup B$), where the chase along definition route causes ascending and descending movements in subsumption relation, the break down of owl:unionOf into the superclasses list in Procedure 6 prevents to happen such infinite cycle calculation. This extended structural subsumption algorithm is incomplete for the existential restriction, but useful as OWL reasoning for most cases in practice.

3.6 Satisfiability Check

The proactive entailment reduces the load of satisfiability check. For example, when programmers attempt to define an object ambiguously (to define an object to a more abstract class), if the domain and range definition is available, then SWCLOS defines an object more specifically (defines an object to a more special class), with fitting the domain and range restriction. Nevertheless, the satisfiability check is useful to prevent programmers from importing bugs into ontologies. We implemented the domain and range checking, value restriction checking, filler restriction checking, cardinality checking, disjoint-pair checking, etc. Additional unsatisfiability rules to the OWL definition are summarized in Table 3 in Appendix.

3.7 OWL Entailment Rules

We note that the complete set of OWL entailment rules are not known, although ter Horst [Horst2004, Horst2005] has been investigating to make them clear. We emphasize that the prover of Tableau Algorithms is insufficient for the proactive entailment. The work of DL prover is to test the membership of individuals and the subsumption of classes. Precisely, it involves satisfiability check of concepts with the refutation. It implies to make a query for the prover. However, in order to perform proactive entailments, we need to sense the situation in which an entailment is deductive, and we must know what query is effective in the situation. In other words, if we know entailment rules, we can set an appropriate query to the prover in the situation, or we can proactively perform the entailment rules by properly applying the rules in the situation, or we can procedurally encode the entailment rules in software tools.

Hereafter in this subsection, many entailment rules in OWL are introduced and discussed how those rules are implemented in SWCLOS. The entailment rules that is denoted **rdfp**** represent P-entailment rules in [Horst2005]. The others denoted **rule*** are published here. See Table 2 in Appendix.

SameAs Group, EquivalentClass Group, EquivalentProperty Group: `sameAs` relation is reflexive (**rdfp6** in [Horst2005]) and transitive (**rdfp7**). So, all related individuals make one group upon `sameAs`. In SWCLOS, the group information that is a collection of related individuals is registered to each individual of group members. The `equivalentClass` is also reflexive (**rdfp12a**) and transitive (**rdfp12c**). Therefore, the same machinery is adopted for `equivalentClass` as `sameAs`. `equivalentProperty` is also the same. Such information is used in the subsumption calculation as explaned in Subsection 3.5.

DifferentFrom Pairs and DisjointWith Pairs: On the other hand, `differentFrom` is reflexive but not transitive. Therefore, the pairwise relation is not resolved into one group. In SWCLOS, the other member of a pair is registered to each individual. This is same for `disjointWith`.

If a class is disjoint with another class, the subclasses of each disjoint superclass are also disjoint each others. See **rule4** in Table 2, which is implemented in function `owl-disjoint-p`. If disjoint classes are specified as multiple classes in an instance definition, SWCLOS signals an alarm of unsatisfiability. See `unsatisfiability3` and 4 in Table 3.

FunctionalProperty: The entailment rule is described by **rdfp1** in [Horst2005]. SWCLOS maintains the bookkeeping of the inverse of `FunctionalProperty`.Then, `owl-same-p` infers this equality on individuals.

InverseFunctionalProperty: The semantics and the entailment rule is just inversely same as `functionalProperty`. See **rdfp2**. `owl-same-p` infers this equality on individuals.

Intersection of Concepts: If $A \equiv C_1 \sqcap \ldots \sqcap C_n$ (where $i = 1, \ldots, n$), then $A \sqsubseteq C_i$. SWCLOS adds every class C_i into the direct-superclasses list of class A from owl:intersectionOf assertions.

Union of Concepts: If $A \equiv C_1 \sqcup \ldots \sqcup C_n$ (where $i = 1, \ldots, n$), then $C_i \sqsubseteq A$. SWCLOS adds class A into the class-superclasses list of every class C_i from owl:unionOf assertions.

Complement Concepts: The complement relation is reflexive (see **rule5**) and the entails the disjointedness (**rule6**). SWCLOS registers one of the pair to both ones for complementness and disjointedness.

3.8 Calculation Efficiency of SWCLOS

The amount of loading time for Food Ontology and Wine Ontology is 2 seconds, and the amount of loading time for Lehigh University Benchmark (LUBM)[10] is 35 seconds for the data of 3235 persons + 659 courses + 6 departments + 759 university in Allegro Common Lisp 8.0 on MS-Windows 2000 with Pentium 4 (CPU 2.6GHz) and 1GB RAM. There is no stress to reply to the LUBM 14 queries for the above loaded data by lisp codes in ordinal programming manner.

4 OWL-Full and Meta-modeling

In this section, we demonstrate with two examples why the meta-class is needed and how it is used for OWL-Full.

[10] http://swat.cse.lehigh.edu/projects/lubm/

4.1 Meta-class for Role and Filler Attachment

Suppose that wine brands are ID-numbered by *International Wine Society*. Since there are mixed together brand wines such as vin:Zinfandel with non-brand wine concepts such as vin:CaliforniaWine in Wine Ontology, we must distinguish them at first. Even if we introduce two new classes as a subclass of vin:Wine, namely BrandWine of which instances have an ID-number and NonBrandWineConcept that does not provide ID-number, we cannot attach an ID-number to wine classes such as vin:Zinfandel (and can attach an ID-number to wine instances such as vin:ElyseZinfandel). Because a brand wine class should be a subclass of BrandWine but should not be an instance of BrandWine. In order to attach a role and filler to a class, a class of the class is required. The solution in SWCLOS is shown below.

```
(defResource BrandWine (rdf:type owl:Class)
  (rdfs:subClassOf vin:Wine owl:Class)) -> #<owl:Class BrandWine>
(defResource NonBrandWineConcept (rdf:type owl:Class)
  (rdfs:subClassOf vin:Wine owl:Class)) -> #<owl:Class NonBrandWineConcept>
(defProperty hasIDNumber (rdf:type owl:ObjectProperty)
  (rdfs:domain BrandWine)
  (rdfs:range xsd:positiveInteger))     -> #<owl:ObjectProperty hasIDNumber>
(defResource vin:Zinfandel (rdf:type BrandWine)
  (hasIDNumber 12345))                  -> #<BrandWine vin:Zinfandel>
(get-form vin:Zinfandel)
 -> (BrandWine vin:Zinfandel (rdf:about #<uri http://www.w3.org/TR ...
  (rdfs:subClassOf (owl:hasValueRestriction ...
                        ...
  (owl:intersectionOf vin:Wine
    (owl:hasValueRestriction ...
    (owl:cardinalityRestriction ...
  (hasIDNumber 12345))
```

4.2 Treatment of Instance as Class

In the OWL-S specification[11] for Semantic Web Services, the range of property `process:hasPrecondition` is `expr:Condition`, and an instance of `expr:Condition` may have a value of `expr:expressionBody`. Suppose that we have many kinds of conditions and need to classify actual conditions to one of these condition classes. For example, we have many operational modes in the rocket launch operation [Misono2005], and each operational mode selects applicable services through preconditions. Note that an `expr:expressionBody` is different each other by operational modes and it identifies each condition class. Here we need to attach `expr:expressionBody` value to class-like preconditions, in order to record actual conditions in operation and store them as instances of each operational conditions. Please recall that `expr:expressionBody` value may be attached to an instance of but cannot be attached to `expr:Condition` per se.

[11] http://www.daml.org/services/owl-s/1.1/

In SWCLOS, the problem is solved as follows. Here gxprocess:Precondition is a metaclass, since it is a subclass of owl:Class. Thus, PipeCoolDownMode-, TankCoolDownMode-, and RocketTankingMode-Precondition turn out classes within the boundary of the schema of OWL-S 1.1.

```
(defResource gxprocess::Precondition (rdf:type owl:Class)
  (rdfs:comment "This is a meta-class for precondition.")
  (rdfs:subClassOf owl:Class expr:Condition))
(defResource gxprocess::OperationModePrecondition
  (rdf:type gxprocess::Precondition)
  (rdfs:label :en "operation mode precondition")
  (rdfs:subClassOf expr:Condition gxdomain::OperationMode)
  (expr:expressionBody ... ))
(defResource gxprocess::PipeCoolDownModePrecondition
  (rdf:type gxprocess::Precondition)
  (rdfs:label :en "pipe cool-down mode precondition")
  (rdfs:subClassOf gxprocess::OperationModePrecondition
                   gxdomain::PipeCoolDownMode)
  (expr:expressionBody ... ))
(defResource gxprocess::TankCoolDownModePrecondition
  (rdf:type gxprocess::Precondition)
  (rdfs:label :en "tank cool-down mode precondition")
  (rdfs:subClassOf gxprocess::OperationModePrecondition
                   gxdomain::TankCoolDownMode)
  (expr:expressionBody ... ))
(defResource gxprocess::RocketTankingModePrecondition
  (rdf:type gxprocess::Precondition)
  (rdfs:label :en "rocket tanking mode precondition")
  (rdfs:subClassOf gxprocess::OperationModePrecondition
                   gxdomain::RocketTankingMode)
  (expr:expressionBody ... ))
```

5 Conclusion

Description Logics provide the means to formalize the application domain, and OWL becomes a modeling language for domain modeling in Software Engineering. SWCLOS is a language for ontology description in OWL, and simultaneously it is an Object-Oriented Programming language on Common Lisp. Therefore, programmers may exchange their idea on software systems on the firm base of Description Logic, and then they can instantiate the formalization and develop working lisp programs on the continuous ground of CLOS. In this paper, we introduced SWCLOS and explained OWL reasoning in SWCLOS. Strictly, the structural subsumption algorithm extended to OWL is still incomplete for the existential restriction, but the system works effectively in most cases of pratical use. SWCLOS provides OWL-Full performance with meta-modeling in CLOS, and we demonstrated some examples in OWL-Full programming. The incompleteness in OWL reasoning caused by the existential restriction will be solved in the future, by introducing First-Order Logic.

Acknowledgment

The most of the work was done as a part of the Japanese IT project entitled 'Building a Support System for the Large-Scale Operation System using Information Technology' under the contract with the Ministry of Education, Culture, Sports, and Technology (MEXT). We thank Prof. Mizoguchi at Osaka University who was a co-researcher in the project and Emeritus Prof. Ohsuga who was the chairperson of the Technology Evaluation Committee of the project. We also appreciate Dr. Kaneiwa, who discussed the entailments of Wine Ontology.

References

[Baarder2003] Baarder, F., W. Nutt: Basic Description Logics, The Description Logic Handbook (eds. Baader et al.). Chap. 2, Cambridge (2003) 43–95

[Borgida2003] Borgida, A., R. J. Brachman: Conceptual Modeling with Description Logics, The Description Logic Handbook (eds. Baader et al.). Chap. 10, Cambridge (2003) 349–372

[Horst2004] Horst, H. J. ter: Extending the RDFS Entailment Lemma. ISWC2004, (2004) 79–91

[Horst2005] Horst, H. J. ter: Combining RDF and Part of OWL with Rules: Semantics, Decidability, Complexity. ISWC2005, (2005) 668–684

[Kaneiwa2005] Kaneiwa, K. and R. Mizoguchi:Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2005), LNCS 3702, Springer–Verlag, (2005) 169–184

[Kiczales1991] Kiczales, G., J. des Rivières, and D. G. Bobrow: The Art of the Metaobject Protocol. MIT Press (1992)

[Koide2004] Koide, S., Kawamura M.: SWCLOS: A Semantic Web Processor on Common Lisp Object System. ISWC2004 Demos, http://iswc2004.semanticweb.org/demos/32/ (2004)

[Koide2005] Koide, S., J. Aasman, S. Haflich: OWL vs. Object Oriented Programming, the 4th International Semantic Web Conference (ISWC 2005), Workshop on Semantic Web Enabled Software Engineering (SWESE), (2005) http://www.mel.nist.gov/msid/conferences/SWESE/repository/8owl-vs-OOP.pdf

[Misono2005] Misono, S., S. Koide, N. Shimada, M. Kawamura, and S. Nagano: Distributed Collaborative Decision Support System for Rocket Launch Operation, IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics (AIM2005), (2005)

[Mizoguchi2004] Mizoguchi, R.:Tutorial on Ontological Engineering - Part 2: Ontology Development, Tools and Languages, New Generation Computing, OhmSha and Springer, **22**-1, (2004) 61–96

[Nardi2003] Nardi, D., R. J. Brackman: An Introduction to Description Logics, The Description Logic Handbook (eds. Baader et al.). Chap. 1, Cambridge (2003) 1–40

[SETF2006] A Semantic Web Primer for Object-Oriented Software Developers, http://www.w3.org/TR/2006/NOTE-sw-oosd-primer-20060309/, W3C (2006)

A OWL Axioms and Entailment Rules

Table 1. Additional OWL Axioms for SWCLOS

axiom1	`Thing subClassOf Resource`
axiom2	`Class subClassOf Thing`
axiom3	`FunctionalProperty type Class`
axiom4	`InverseFunctionalProperty type Class`
axiom5	`FunctionalProperty disjointWith InverseFunctionalProperty`

Table 2. Entailment Rules in OWL for SWCLOS

	If	Then
rule0	r `type Restriction`	r `subtype Resource`
rule1a	$v\ p\ w$	v `subtype Thing`
rule1b	$v\ p\ w$	w `subtype Thing`
rule2a	u `intersectionOf` $\{v_j \ldots\}$	v_j `type Class`
rule2b	u `unionOf` $\{v_j \ldots\}$	v_j `type Class`
rule3	x `distinctMembers` $\{x_j \ldots\}$	x_j `type Thing`
rule4	u `disjointWith` v	
	u' `subClassOf` u	
	v' `subClassOf` v	u' `disjointWith` v'
rule5	u `complementOf` v	v `complementOf` u
rule6	u `complementOf` v	v `disjointWith` u
rule7	u `oneOf` $\{x_j \ldots\}$	x_j `type` u
rule8	v `allValuesFrom` w	
	v `onProperty` p	
	p `range` u	w `subtype` u

Table 3. Unsatisfiability in OWL for SWCLOS

	Unsatisfiable Conditions
unsatisfiability1	u `oneOf` $\{ x_i \ldots \}$
	y `type` u
	y `differentFrom` x_i
unsatisfiability2	x `differentFrom` y
	x `sameAs` y
unsatisfiability3	u `disjointWith` v
	v `equivalentOf` u
unsatisfiability4	u `disjointWith` v
	x `type` u
	x `type` v

Visualizing Defeasible Logic Rules for the Semantic Web

Efstratios Kontopoulos[1], Nick Bassiliades[1], and Grigoris Antoniou[2]

[1] Department of Informatics, Aristotle University of Thessaloniki
GR-54124 Thessaloniki, Greece
{nbassili, skontopo}@csd.auth.gr
[2] Institute of Computer Science, FO.R.T.H.
P.O. Box 1385, GR-71110, Heraklion, Greece
antoniou@ics.forth.gr

Abstract. Defeasible reasoning is a rule-based approach for efficient reasoning with incomplete and conflicting information. Such reasoning is useful in many Semantic Web applications, like policies, business rules, brokering, bargaining and agent negotiations. Nevertheless, defeasible logic is based on solid mathematical formulations and is, thus, not fully comprehensible by end users, who often need graphical trace and explanation mechanisms for the derived conclusions. Directed graphs can assist in confronting this drawback. They are a powerful and flexible tool of information visualization, offering a convenient and comprehensible way of representing relationships between entities. Their applicability, however, is balanced by the fact that it is difficult to associate data of a variety of types with the nodes and the arcs in the graph. In this paper we try to utilize digraphs in the graphical representation of defeasible rules, by exploiting the expressiveness and comprehensibility they offer, but also trying to leverage their major disadvantage, by defining two distinct node types, for rules and atomic formulas, and four distinct connection types for each rule type in defeasible logic and for superiority relationships. The paper also briefly presents a tool that implements this representation methodology.

1 Introduction

Defeasible reasoning [22], a member of the non-monotonic reasoning family, constitutes a simple rule-based approach to reasoning with incomplete and conflicting information. This approach offers two main advantages: (a) enhanced representational capabilities, allowing one to reason with incomplete and contradictory information, coupled with (b) low computational complexity compared to mainstream non-monotonic reasoning. Defeasible reasoning can represent facts, rules as well as priorities and conflicts among rules. Such conflicts arise, among others, from rules with exceptions, which are a natural representation for policies and business rules [2]. And priority information is often available to resolve conflicts among rules. Potential applications include security policies ([19]), business rules [1], e-contracting [15], personalization, brokering [5], bargaining and agent negotiations ([14]).

However, although defeasible reasoning features a significant degree of expressiveness and intuitiveness, it is still based on a solid mathematical formulation, which, in many cases, may seem too complicated. So, end users might often consider the

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 278 – 292, 2006.
© Springer-Verlag Berlin Heidelberg 2006

conclusion of a defeasible logic theory incomprehensible and complex and, thus, a graphical trace and an explanation mechanism would be very beneficial.

In this paper we try to utilize directed graphs in the graphical representation of defeasible rules. Directed graphs, or digraphs, as they are usually referred to, are a special case of graphs that constitute a powerful and convenient way of representing relationships between entities [11].

Usually in a digraph, entities are represented as nodes and relationships as directed lines or arrows that connect the nodes. Each arrow connects only two entities at a time and there can be no two (or more) arrows that connect the same pair of nodes. The orientation of the arrows follows the flow of information in the digraph. A mathematical definition of directed graphs as well as details on graph theory in general can be found in [13].

Digraphs offer a number of advantages to information visualization:

- *comprehensibility*: the information that a digraph contains can be easily and accurately understood by humans [21] and
- *expressiveness*: although the appearance and structure of a digraph may seem simplistic, its topology bears non-trivial information [11]

Furthermore, in the case of graphical representation of logic rules, digraphs seem to be extremely appropriate, since they offer a number of extra advantages:

- explanation of derived conclusions: the series of inference steps in the graph can be easily detected and retraced [3]
- proof visualization and validation: by going backwards from the conclusion to the triggering conditions, one can validate the truth of the inference result
- especially in the case of defeasible logic rules, the notion of direction can also assist in graphical representations of rule attacks, superiorities etc.

However, their major disadvantage is the fact that it is difficult to associate data of a variety of types with the nodes and with the connections between the nodes in the graph [11].

Therefore, in this paper we attempt to exploit the expressiveness and comprehensibility of directed graphs, as well as their suitability for rule representation, but also try to leverage their aforementioned disadvantage, by adopting a new, “enhanced” digraph approach. This visualization scheme was implemented as part of the VDR-DEVICE system, which is a visual integrated development environment for developing and using defeasible logic rule bases on top of RDF ontologies [7] for the Semantic Web and is also briefly presented in this work.

The rest of this paper is organized as follows: Section 2 explains an approach of rule representation with digraphs, including representation of literals, arguments, variables and simple conditions. Section 3 introduces the semantics of defeasible logics and presents the representation methodology of defeasible reasoning with directed graphs. Section 4 briefly presents the architecture and functionality of the VDR-DEVICE system, while the next section describes representational enhancements, based on the VDR-DEVICE object-oriented model, including the recently added utility for drawing directed defeasible rule graphs. Finally, section 6 discusses related work, followed by conclusions and directions for future work.

2 Representing Rules with Digraphs

In an attempt to leverage the most important disadvantage of graphs (inability to use a variety of distinct entity types), the digraphs in our approach will contain two kinds of nodes, similarly to the methodology followed by [18]. The two node types will be:

- literals, represented by rectangles, which we call "*literal boxes*"
- rules, represented by circles

Thus, according to this principle, the following rule base:

```
p: if A then B
q: if B then ¬C
```

can be represented by the directed graph shown in Fig. 1.

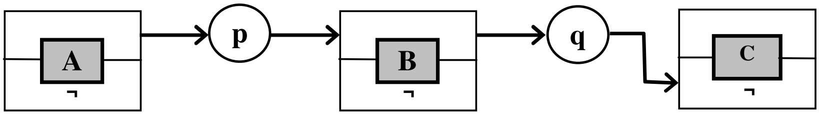


Fig. 1. Digraph featuring a conjunction

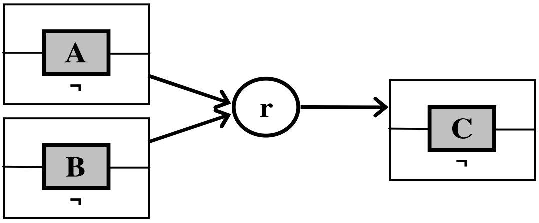


Fig. 2. Digraph featuring a conjunction

Each literal box consists of two adjacent "*atomic formula boxes*", with the upper one of them representing a positive atomic formula and the lower one representing a negated atomic formula. This way, the atomic formulas are depicted together clearly and separately, maintaining their independence.

In the case of a rule body that consists of a conjunction of literals (`if ¬A and B then C`) the representation is not profoundly affected, as illustrated in Fig. 2:

2.1 Representing Arguments and Conditions

So far we have demonstrated how rules are represented by interconnecting literal boxes with rule nodes. However, we have not included how literal arguments are presented, either being variables or constants. Furthermore, variables are usually associated with simple conditions, such as $X > 4$, which theoretically could be represented as predicates, but practically it is more convenient to consider them more closely related to the closest literal that contains the corresponding variable as an argument.

Arguments can be incorporated inside the literal box, just after the predicate name of each literal box. We call the set of all arguments for each literal box, an *argument pattern*. For example, the literal a(X,2) is represented as in Fig. 3. Simple conditions associated with any of the variables of a literal can also appear inside the literal box. However, since there can be many conditions, each one of them appears on a separate line (*condition pattern*) below the literal. For example, if the fragment a(X,Y),Y>4 appears in a rule condition, it can be represented as in Fig. 4.

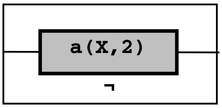


Fig. 3. Representing arguments of literals

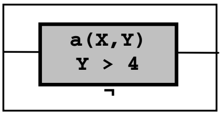


Fig. 4. Representing simple conditions on variables

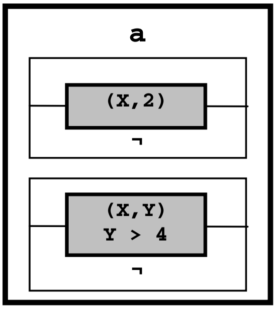


Fig. 5. Predicate box and predicate patterns

A certain predicate, say a, can appear many times in a rule base, in many rule conditions or even rule conclusions (if it is not a base predicate, i.e. a fact). We would like to group all literal boxes of the same predicate so that the user can visually comprehend that all such literal boxes refer to the similar set of literals. In order to achieve this, we introduce the notion of a *predicate box*, which is simply a container for all the literal boxes that refer to the same predicate. Predicate boxes are labeled with the name of the predicate. Furthermore, the literal boxes contained inside the predicate box "lose" the predicate name, since the latter is located at the top of the predicate box. Such literal boxes, which appear inside predicate boxes and express conditions on instances of the specific predicate extension, are called *predicate patterns*.

For example, the literal boxes of Fig. 3 and Fig. 4 can be grouped inside a pattern box as in Fig. 5. Notice that each predicate pattern contains exactly one argument pattern and zero, one or more condition patterns.

3 Defeasible Logics and Digraphs

As can be observed from the previous section, digraphs "enhanced" with the addition of distinct node types, offer a significant level of expressiveness in representing rules. The next step is to use directed graphs in the representation of defeasible logic rules, which are more demanding in representational capabilities.

A *defeasible theory D* (i.e. a knowledge base or a program in defeasible logic) consists of three basic ingredients: a set of facts (F), a set of rules (R) and a superiority relationship (>). Therefore, D can be represented by the triple (F, R, >).

In defeasible logic, there are three distinct types of rules: strict rules, defeasible rules and defeaters. In our approach, each one of the three rule types will be mapped to one of three distinct connection types (i.e. arrows), so that rules of different types can be represented clearly and distinctively.

So, the first rule type in defeasible reasoning is *strict rules*, which are denoted by $A \rightarrow p$ and are interpreted in the typical sense: whenever the premises are indisputable, then so is the conclusion. An example of a strict rule is: "*Penguins are birds*", which would become: `r`$_1$`: penguin(X) →bird(X)`, and, using a digraph, this would be represented by Fig. 6.

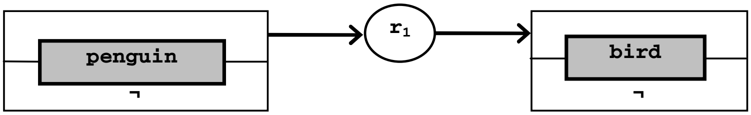


Fig. 6. Visual representation of a strict rule

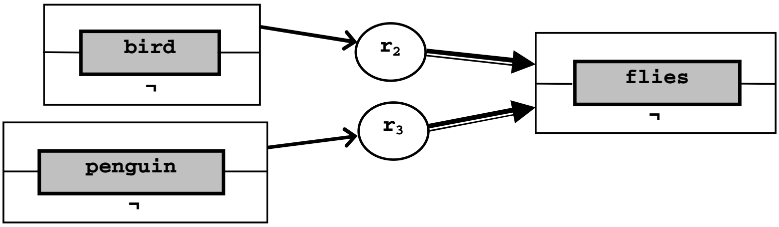


Fig. 7. Visual representation of defeasible rules

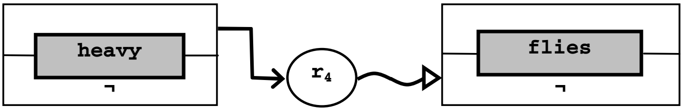


Fig. 8. Visual representation of a defeater

Notice that in the rule graph we only represent the predicate and not the literal (i.e. predicate plus all the arguments) because we are mainly interested in making clear to the user the interrelationships between the concepts (through the rules) and not the complete details of the defeasible theory.

Contrary to strict rules, *defeasible rules* can be defeated by contrary evidence and are denoted by *A p*. Examples of defeasible rules are r_2: bird(X) flies(X), which reads as: "*Birds typically fly*" and r_3: penguin(X) ¬flies(X), namely: "*Penguins typically do not fly*". Rules r_2 and r_3 would be mapped to the directed graphs of Fig. 7.

Defeaters, denoted by *A ~> p*, are rules that do not actively support conclusions, but can only prevent some of them. In other words, they are used to defeat some defeasible conclusions by producing evidence to the contrary. An example of such a defeater is: r_4: heavy(X) ~> ¬flies(X), which reads as: "*Heavy things cannot fly*". This defeater can defeat the (defeasible) rule r_2 mentioned above and it can be represented by Fig. 8.

Finally, the *superiority relationship* among the rule set R is an acyclic relation > on R, that is, the transitive closure of > is irreflexive. Superiority relationships are used, in order to resolve conflicts among rules. For example, given the defeasible rules r_2 and r_3, no conclusive decision can be made about whether a penguin can fly or not, because rules r_2 and r_3 contradict each other. But if the superiority relationship $r_3 > r_2$ is introduced, then r_3 overrides r_2 and we can indeed conclude that the penguin cannot fly. In this case rule r_3 is called *superior* to r_2 and r_2 *inferior* to r_3. In the case of superiority relationships a fourth connection type is introduced. Thus, the aforementioned superiority relationship would be represented by Fig. 9.

The set of rules mentioned in this section, namely rules r_1, r_2, r_3 and r_4, form a bigger directed rule graph, which is depicted in Fig. 10.

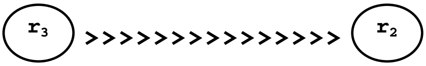


Fig. 9. Visual representation of a superiority relation

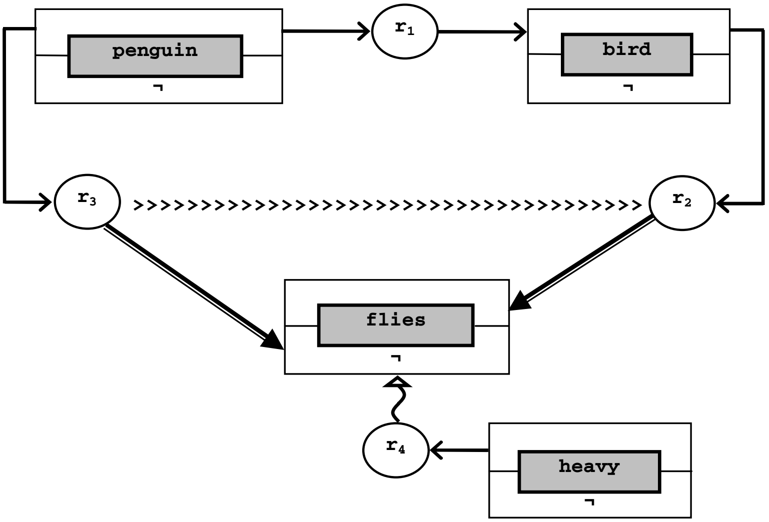


Fig. 10. The digraph formed by the rules r_1, r_2, r_3 and r_4

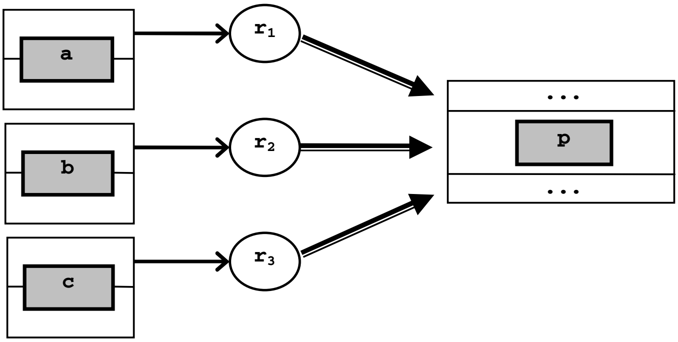


Fig. 11. Representation of conflicting literals as a digraph

Finally, another important type of conflicting evidence in defeasible reasoning is the notion of *conflicting literals*. In some applications, e.g. making an offer in a price negotiation setting, literals are often considered to be conflicting and at most one of a certain set should be derived. Consider the following three rules, which all produce the same literal type as a conclusion, and the constraint that requires at most one of the literals to be true:

r_1: a(X) p(X), r_2: b(X) p(X), r_3: c(X) p(X)

p(X),p(Y),X≠Y -> ⊥

The graph drawn by these rules is depicted in Fig. 11 and, as can be observed, all three rules produce the same result type, which is included in a *single literal truth box*. Of course, superiority relationships could still determine the priorities among the rules.

4 The VDR-DEVICE System

VDR-DEVICE is a visual, integrated development environment for developing and using defeasible logic rule bases on top of RDF ontologies [7]. It consists of two primary components:

1. DR-DEVICE, the reasoning system that performs the RDF processing and inference and produces the results, and
2. DRREd (Defeasible Reasoning Rule Editor), the rule editor, which serves both as a rule authoring tool and as a graphical shell for the core reasoning system.

4.1 The Reasoning System - Architecture and Functionality

The core reasoning system of VDR-DEVICE is DR-DEVICE [6] and consists of two primary components (Fig. 12): The *RDF loader/translator* and the *rule loader/translator*. The user can either develop a rule base (program, written in the RuleML-like syntax of VDR-DEVICE – see for a fragment) with the help of the rule editor described in the following sections, or he/she can load an already existing one, probably developed manually. The rule base contains: (a) a set of rules, (b) the

URL(s) of the RDF input document(s), which is forwarded to the RDF loader, (c) the names of the derived classes to be exported as results and (d) the name of the RDF output document.

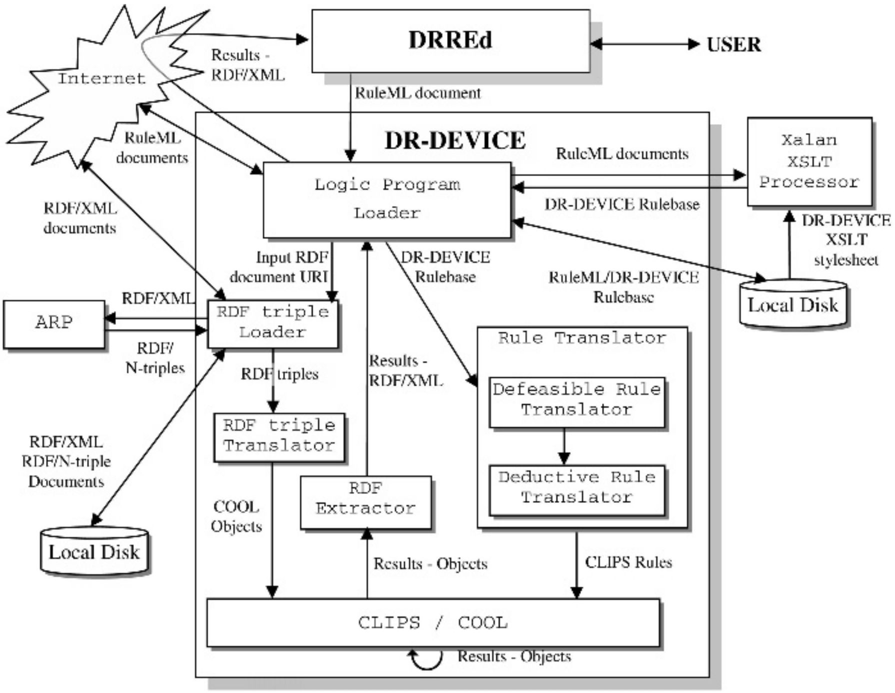


Fig. 12. The VDR-DEVICE system architecture

```
<imp>
  <_rlab ruleID="r1" ruletype="strictrule">
    <ind>r1</ind>
  </_rlab>
  <_head>
    <atom>
      <_opr> <rel>bird</rel> </_opr>
      <_slot name="name"> <var>X</var> </_slot>
    </atom>
  </_head>
  <_body>
    <atom>
      <_opr> <rel>penguin</rel> </_opr>
      <_slot name="name"> <var>X</var> </_slot>
    </atom>
  </_body>
</imp>
```

Fig. 13. A strict rule, written in the RuleML-compatible language of DR-DEVICE (this fragment displays rule r_1 of section 3)

The rule base is then submitted to the *rule loader* which transforms it into the native CLIPS-like syntax through an XSLT stylesheet and the resulting program is then forwarded to the *rule translator*, where the defeasible logic rules are compiled into a set of CLIPS production rules [12]. This is a two-step process: First, the defeasible logic rules are translated into sets of deductive, derived attribute and aggregate attribute rules of the basic deductive rule language, using the translation scheme described in [6]. Then, all these deductive rules are translated into CLIPS production rules according to the rule translation scheme in [8].

Meanwhile, the *RDF loader* downloads the input RDF documents, including their schemas, and translates RDF descriptions into CLIPS objects [12], according to the RDF-to-object translation scheme in [8], which is briefly described below.

The inference engine of CLIPS performs the reasoning by running the production rules and generates the objects that constitute the result of the initial rule program. The compilation phase guarantees correctness of the reasoning process according to the operational semantics of defeasible logic. Finally, the result-objects are exported to the user as an RDF/XML document through the RDF extractor. The RDF document includes the instances of the exported derived classes, which have been proved.

The Object-Oriented RDF Data Model

The DR-DEVICE system employs an OO RDF data model, which is different from the established triple-based data model for RDF. The main difference is that DR-DEVICE treats properties both as first-class objects and as normal encapsulated attributes of resource objects. In this way properties of resources are not scattered across several triples as in most other RDF inferencing systems, resulting in increased query performance due to less joins. For example, Fig. 14 shows an RDF resource that describes a person, which is represented as a COOL object in DR-DEVICE (Fig. 15).

```
<rdf:RDF ... xmlns:ex="http://...rdfs" xmlns:ex_in="http://...rdf">
 <ex:person rdf:about="&ex_in;p1">
   <ex:name>John Smith</ex:name>
   <ex:age rdf:datatype="&xsd;integer">25</ex:age>
   <ex:sex>M</ex:sex>
 </ex:person>
 ...
</rdf:RDF>
```

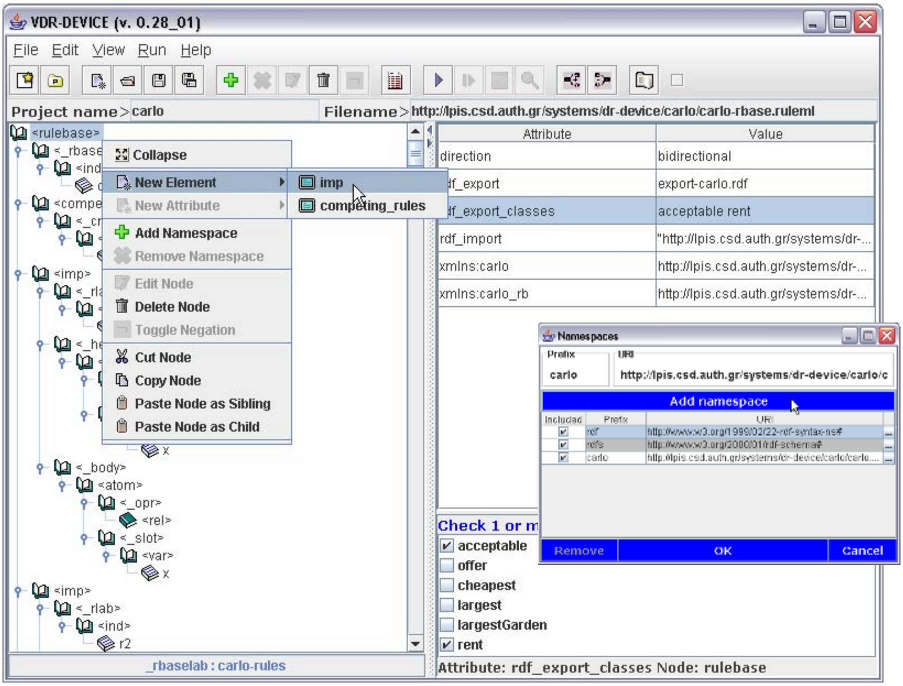
Fig. 14. RDF document excerpt for a person

```
[ex_in:p1] of ex:person
(ex:name "John Smith")
(ex:age 25)
(ex:sex "M")
```

Fig. 15. COOL object for the RDF resource of Fig. 14

4.2 Rule Editor – Design and Functionality

Writing rules in RuleML can often be a highly cumbersome task. Thus, the need for authoring tools that assist end-users in writing and expressing rules is apparently imperative. VDR-DEVICE is equipped with DRREd (Fig. 16), a Java-built visual rule editor that aims at enhancing user-friendliness and efficiency during the development of VDR-DEVICE RuleML documents [7]. Its implementation is oriented towards simplicity of use and familiarity of interface. Other key features of the software include: (a) functional flexibility - program utilities can be triggered via a variety of overhead menu actions, keyboard shortcuts or popup menus, (b) improved development speed - rule bases can be developed in just a few steps and (c) powerful safety mechanisms – the correct syntax is ensured and the user is protected from syntactic or RDF Schema related semantic errors.



Project name>carlo Filename>http://lpis.csd.auth.gr/systems/dr-device/carlo/carlo-rbase.ruleml

Attribute	Value
direction	bidirectional
df_export	export-carlo.rdf
df_export_classes	acceptable rent
rdf_import	"http://lpis.csd.auth.gr/systems/dr-...
xmlns:carlo	http://lpis.csd.auth.gr/systems/dr-...
xmlns:carlo_rb	http://lpis.csd.auth.gr/systems/dr-...

Prefix: carlo URI: http://lpis.csd.auth.gr/systems/dr-device/carlo/c

Included	Prefix	URI
☑	rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
☑	rdfs	http://www.w3.org/2000/01/rdf-schema#
☑	carlo	http://lpis.csd.auth.gr/systems/dr-device/carlo/carlo....

Check 1 or m
☑ acceptable
□ offer
□ cheapest
□ largest
□ largestGarden
☑ rent

Attribute: rdf_export_classes Node: rulebase

Fig. 16. The graphical rule editor and the namespace dialog window

The rule base is displayed in XML-tree format, which is one of the most intuitive means of displaying RuleML-like syntax, because of its hierarchical nature. The user has the option of navigating through the entire tree and can add to or remove elements from the tree. However, since each rule base is backed by a DTD document, potential addition or removal of tree elements has to obey to the DTD limitations. Therefore, the rule editor allows a limited number of operations performed on each element, according to the element's meaning within the rule tree.

By selecting an element from the tree, the corresponding attributes are displayed each time. The user can also perform editing functions on the attributes, by altering the value for each one of them. However, the values that the user can insert are obviously limited by the chosen attribute each time.

5 Representing the Object Model of VDR-DEVICE

VDR-DEVICE adopts a purely object-oriented model for RDF, encapsulating the properties as attributes in classes. The rule bases developed by DRREd are, thus, compelled to follow this principle. As a consequence, the graphical representation of a VDR-DEVICE rule base, following the methodology described in previous sections (sections 2 and 3) has to be modified, in order to closely reflect the associations between classes and atoms and between arguments and properties.

5.1 Class Boxes, Class Patterns and Slot Patterns

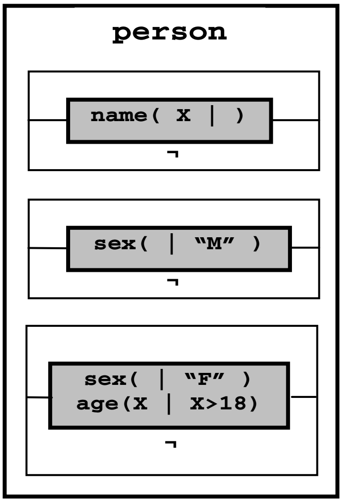
For every class in the rule base (i.e. classes that lie at rule bodies and heads) a *class box* is constructed, which is simply a container. Class boxes are the equivalent of *predicate boxes*. The class boxes are populated with one or more *class patterns* during the development of the rule base. Class patterns are the equivalent of *predicate patterns*. For each atomic formula inside a rule head or body, a new class pattern is created and is associated with the corresponding class box. In practice, class patterns express conditions on instances of the specific class.

Visually, class patterns appear as literal boxes, whose design was thoroughly described in section 2. The mapping of class patterns to literal boxes is justified by the fact that RuleML atoms are actually atomic formulas (i.e. they correspond to queries over RDF resources of a certain class with certain property values). As a result, the truth value associated with each returned class instance will be either positive or negative.

Class patterns are populated with one or more *slot patterns*. Each slot pattern consists of a slot name and, optionally, a variable and a list of value constraints. The variable is used in order for the slot value to be unified, with the latter having to satisfy the list of constraints. In other words, slot patterns represent conditions on slots (or class properties).

Slot patterns are supposed to be the equivalent of *argument patterns* and *condition patterns*. However, there are certain differences that arise from the different nature of the tuple-based model of predicate logic and the object-based model of VDR-DEVICE. In VDR-DEVICE class instances are queried via named slots rather than positional arguments. Not every slot needs to be queried and slot position inside the object is irrelevant. Therefore, instead of a single-line argument pattern we have a set of slot patterns in many lines; each slot pattern is identified by the slot name. Furthermore, in the RuleML syntax of VDR-DEVICE, simple conditions are not attached to the slot patterns; this is reflected to the visual representation where condition patterns are encapsulated inside the associated slot patterns.

An example of all the above can be seen in Fig. 17. The figure illustrates a class box that contains three class patterns applied on the *person* class (see also Fig. 14 and Fig. 15) and a code fragment matching the third class pattern, written in the RuleML-like syntax of VDR-DEVICE. The first two class patterns contain one slot pattern each, while the third one contains two slot patterns. As can be observed, the argument list of each slot pattern is divided into two parts, separated by ”|”; on the left all the variables are placed and on the right all the corresponding expressions and conditions, regarding the variables on the left. In the case of constant values, only the right-hand side is utilized; thus, the second class pattern of the box in Fig. 17, for example, refers to all the *male* persons. This way the content of the slot arguments is clearly depicted and easily comprehended.



```
<atom>
  <_opr>
    <rel href="person"/>
  </_opr>
  <_slot name="sex">
    <ind>"F"</ind>
  </_slot>
  <_slot name="age">
    <_and>
      <var>x</var>
      <function_call
       name="&gt;">
        <var>x</var>
        <ind>18</ind>
      </function_call>
    </_and>
  </_slot>
</atom>
```

Fig. 17. A class box example and a code fragment for the third class pattern

5.2 The Digraph Utility of VDR-DEVICE

DRREd is equipped with a utility that allows the creation of a directed rule graph from the defeasible rule base developed by the editor. This way, users are offered an extra means of visualizing the rule base, besides XML-tree format and, thus, possess a better aspect of the rule base displayed and the inference results produced. Note, however, that the implemented module currently employs the methodology described in sections 2 (excluding section 2.1) and 3 and not the principles described in the previous section (section 5.1). However, this is included in the directions for future work.

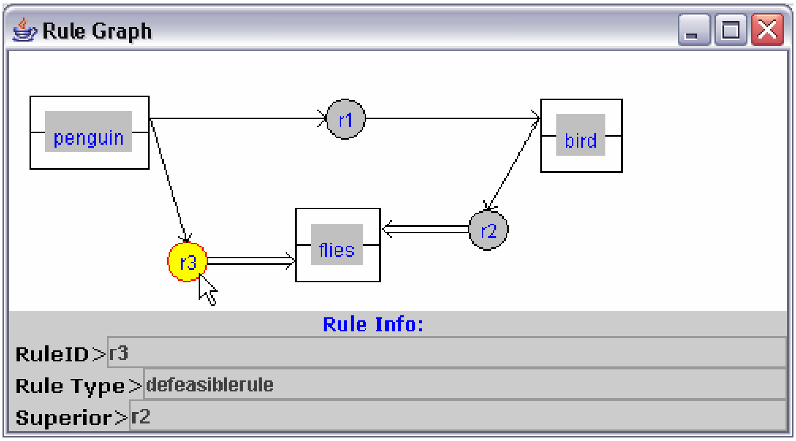
The two aspects of the rule base, namely the XML-tree and the directed graph are correlated, meaning that traversal and alterations in one will be reflected in the other and vice versa. So, if for example the user focuses on a specific element in the tree and then switches to the digraph view, the corresponding element in the digraph will also be selected and the information relevant to it displayed.

More specifically, the digraph drawing utility simply analyzes the rule base into a set of rules, detecting the head and body for every rule, as well as information relevant to each rule (i.e. rule type, ruleID, superiority relationships between rules etc.). Each rule body can potentially consist of more than one atomic formula.

When this analysis comes to an end, the corresponding digraph can be derived, with the more complex rules being placed first (rules with many atomic formulas in the rule body) and the simpler rules being gradually placed afterwards. Since facts can be considered as very simple rules (rules with no body), they are placed into the rule graph in the end.

In case the derived digraph is too big to fit the screen, the user has the option of focusing on a specific part of it and can also traverse the rest of the graph parts by using the scroll controls. Furthermore, similarly to the XML-tree format of the rule base, in the digraph there is also the possibility to collapse or expand certain parts of it. This way, a twofold advantage is offered: (a) the complexity of the digraph is minimized, since only a limited number of graph parts are visible at a time and (b) the level of comprehensibility on behalf of the user is raised, since he/she does not have to focus on the whole graph, but only to a part of it.

Fig. 18 displays the directed rule graph that contains rules r_1, r_2 and r_3 of section 3, produced by the graph drawing utility of DRREd. All nodes can be moved and selected. Selecting a node results in the utility displaying the corresponding attributes.



Rule Info:

RuleID>r3

Rule Type>defeasiblerule

Superior>r2

Fig. 18. The rule graph drawing utility of DRREd

6 Related Work

There exist systems that implement rule representation/visualization with graphs, although we haven't come across a system that represents defeasible logic rules yet. Such an implementation is the Prolog compiler system WIN-PROLOG from LPA [20], which, besides simple rule representation, also offers rule execution tracing. The graphs produced, however, feature an elementary level of detail and, therefore, do not assist significantly in the visualization of the rule bases developed.

Certain knowledge-based system development tools also feature rule and execution graph-drawing. An example is KEE (Knowledge Engineering Environment) [17] that offers several execution control mechanisms. The main features of the software include: (a) a knowledge base development tool, (b) utilities for the interface with the user and (c) graph drawing tools for the knowledge base and execution.

Another example is Graphviz [16], which is an open-source graph visualization software, with several main graph layout programs. Its applications are not limited to drawing rule graphs, but can also include other domains, like software engineering, database and web design, networking and visual interfaces. As a general-purpose graph drawing utility, Graphviz can be applied in rule graph drawing, since it offers variation in graph node types, but does not feature variety in the connection types in the graph and is, therefore, unsuitable for the representation of defeasible rules.

Finally, there have been attempts of creating rule graphs for certain rule types, like association rules ([10], [23]) or production rules [18], but they remained at an elementary stage of development.

7 Conclusions and Future Work

In this paper we argued that graphs can be a helpful tool in the field of information visualization. Especially in the case of rules, directed graphs can be particularly useful, since by definition they embrace the idea of information flow, a notion that is also encountered in rules and inference. Directed graphs have, however, a major disadvantage, which is their inability to associate data of a variety of types with the nodes and with the connections between the nodes in the graph. In this paper we propose an approach that aims at leveraging this disadvantage by allowing different node and connection types in the graph. We also demonstrated that digraphs, “enhanced” with these extra features, can assist significantly in representing defeasible logic rules.

Our future plans involve improvement of the VDR-DEVICE graph-drawing utility demonstrated, by introducing the full representational methodology described in this paper. The issue of scalability (i.e. applicability to more complex rule sets) has to be addressed as well, since the tool currently only deals with simpler rule bases. Also, a user evaluation of the tool should also be considered, in order to realize at what extend the proposed representation is actually helpful. Furthermore, in the future, we plan to delve deeper into the visualization of the proof layer of the Semantic Web architecture by enhancing the rule representation utility with rule execution tracing, explanation, proof visualization, etc. These facilities would be useful in order to automate proof exchange and trust among agents in the Semantic Web and, ultimately, to increase the trust of users towards the Semantic Web.

Acknowledgments

This work was partially supported by the European Network of Excellence REWERSE.

References

[1] Antoniou G. and Arief M., “Executable Declarative Business rules and their use in Electronic Commerce”, *Proc. ACM Symposium on Applied Computing*, 2002.

[2] Antoniou G., Billington D. and Maher M.J., “On the analysis of regulations using defeasible rules”, *Proc. 32nd Hawaii International Conference on Systems Science*, 1999.

[3] Antoniou G., Harmelen F. van, *A Semantic Web Primer*, MIT Press, 2004.

[4] Antoniou G., *Nonmonotonic Reasoning*, MIT Press, 1997.

[5] Antoniou G., Skylogiannis T., Bikakis A., Bassiliades N., “DR-BROKERING – A Defeasible Logic-Based System for Semantic Brokering”, *IEEE Int. Conf. on E-Technology, E-Commerce and E-Service*, pp. 414-417, Hong Kong, 2005.

[6] Bassiliades N., Antoniou, G., Vlahavas I., “A Defeasible Logic Reasoner for the Semantic Web”, *Int. Journal on Semantic Web and Information Systems,* 2(1), pp. 1-41, 2006.

[7] Bassiliades N., Kontopoulos E., Antoniou G., “A Visual Environment for Developing Defeasible Rule Bases for the Semantic Web”, *Proc. International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML-2005)*, A. Adi, S. Stoutenburg, S. Tabet (Ed.), Springer-Verlag, LNCS 3791, pp. 172-186, Galway, Ireland, 2005.

[8] Bassiliades N., Vlahavas I., “R-DEVICE: An Object-Oriented Knowledge Base System for RDF Metadata”, *Int. Journal on Semantic Web and Information Systems*, 2(2), pp. 24-90, 2006.

[9] Boley H., Tabet S., *The Rule Markup Initiative*, www.ruleml.org/

[10] Chakravarthy S., Zhang H., “Visualization of association rules over relational DBMSs”, *Proc. 2003 ACM Symp. on Applied Computing*, ACM Press, pp. 922-926, Melbourne, Florida, 2003.

[11] Clarke D., “An Augmented Directed Graph Base for Application Development”, *Proc. 20th Annual Southeast Regional Conf.*, ACM Press, pp. 155-159, Knoxville, Tennessee, USA, 1982.

[12] *CLIPS Basic Programming Guide* (v. 6.21), www.ghg.net/clips/CLIPS.html.

[13] Diestel R., *Graph Theory (Graduate Texts in Mathematics)*, 2nd ed., Springer, 2000.

[14] Governatori G., Dumas M., Hofstede A. ter and Oaks P., “A formal approach to protocols and strategies for (legal) negotiation”, *Proc. ICAIL 2001*, pp. 168-177, 2001.

[15] Governatori, G., "Representing business contracts in RuleML", *International Journal of Cooperative Information Systems*, 14 (2-3), pp. 181-216, 2005.

[16] Graphviz - Graph Visualization Software, http://www.graphviz.org.

[17] Intellicorp, *The Knowledge Engineering Environment*, 1984.

[18] Jantzen J., “Inference Planning Using Digraphs and Boolean Arrays”, Proc. International Conference on APL, ACM Press, pp. 200-204, New York, USA, 1989.

[19] Li N., Grosof B. N. and Feigenbaum J., “Delegation Logic: A Logic-based Approach to Distributed Authorization”, *ACM Trans. on Information Systems Security*, 6(1), 2003.

[20] LPA WIN-PROLOG, http://www.lpa.co.uk/win.htm.

[21] Nascimento H. A. D. do, “A Framework for Human-Computer Interaction in Directed Graph Drawing”, *Proc. Australian Symp. on Information Visualization*, Australian Computer Society, pp. 63-69, Sydney, Australia, 2001.

[22] Nute D., “Defeasible Reasoning”, *Proc. 20th Int. Conference on Systems Science*, IEEE Press, 1987, pp. 470-477.

[23] Wong P. C., Whitney P., Thomas J., “Visualizing Association Rules for Text Mining”, *Proc. IEEE Symp. on Information Visualization*, IEEE Computer Society, p. 120, 1999.

A Reasoning Algorithm for pD*

Huiying Li[1], Yanbing Wang[1], Yuzhong Qu[1], and Jeff Z. Pan[2]

[1] Department of Computer Science and Engineering, Southeast University, Nanjing
210096, P.R. China
{huiyingli, ybwang, yzqu}@seu.edu.cn
[2] Department of Computing Science, The University of Aberdeen, UK
jpan@csd.abdn.ac.uk

Abstract. pD* semantics extends the 'if-semantics' of RDFS to a subset of the OWL vocabulary. It leads to simple entailment rules and relatively low computational complexity for reasoning. In this paper, we propose a forward-chaining reasoning algorithm to support RDFS entailments under the pD* semantics. This algorithm extends the Sesame algorithm to cope with the pD* entailments. In particular, an optimization to the dependent table between entailment rules is presented to eliminate much redundant inferring steps. Finally, some test results are given to illustrate the correctness and performance of this algorithm.

1 Introduction

RDF (together with RDF Schema, or RDFS) and OWL are important Semantic Web (SW) standards from W3C. Herman Horst [6] proposes a semantic extension of RDFS, called pD* semantics, that supports datatypes, some OWL constructors and axioms. Interestingly, the pD* semantics is in line with the 'if-semantics' of RDFS and weaker than the 'iff-semantics' of OWL. With the 'if-semantics', pD* entailment is NP-complete, which is the same as RDFS entailment. In this paper, we propose a forward-chaining reasoning algorithm for pD* entailment. This algorithm extends the Sesame algorithm and performs an optimization to the dependent table between entailment rules to eliminate much redundant inferring steps. We use the W3C recommended benchmark to evaluate our algorithm - the test results show that the data loading time of this algorithm is better than usual exhaustive forward-chaining algorithms. In addition, we also provide an efficient approximate algorithm for users who want some correct results rapidly but do not require the quick answers to be complete.

The rest of the paper is organized as follows. In section 2, we give a short introduction to the related work. Section 3 gives an overview of the preliminaries about pD* semantics. In section 4 we introduce our forward-chaining reasoning algorithm for pD*. Section 5 discusses our test results. Finally we provide our conclusions in section 6.

2 Related Work

The advent of RDFS represented an early attempt at a SW ontology language based on RDF. As the constructors and axioms provided by RDFS are primi-

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 293–299, 2006.
© Springer-Verlag Berlin Heidelberg 2006

tive, W3C recommends the SW ontology language OWL. There are two kinds of semantics related to RDFS and OWL, namely RDF MT and RDFS(FA) [7]. Accordingly, reasoners support the reasoning of RDFS and OWL can be divided into two categories. Reasoners in the first category, such as Sesame [1] and Jena, are based on the RDFS entailment rules and the extended entailment rules defined in RDF MT. Reasoners in the second category, such as FaCT++ [10], RACER [4] and Pellet [9], support the bottom two layers of the RDFS(FA) semantics [7]. They usually implement tableau-based decision procedures for Description Logics (DLs).

Our work discussed in this paper belongs to the first category. We extend the Sesame algorithm to cope with not only RDFS entailment rules but also pD* entailment rules and optimize the dependencies between rules for eliminating much redundant inferring steps.

3 pD* Semantics

It is well known that extending RDF with the OWL constructors and axioms (i.e. OWL Full) would lead to undecidability. It has also been shown in [8] that extending RDF MT to First Order Logic (FOL) results in a collapse of the model theory. So far there are two solutions to providing decidable extensions for RDF: we can adopt either the FA semantics [7], which has been shown to be compatible with OWL DL, or the pD* semantics [6], which extends RDF MT to cover some OWL constructors and axioms.

Interestingly, the pD* semantics is in line with the 'if-semantics' of RDFS and weaker than the 'iff-semantics' that is used in the RDF-compatible semantic for OWL DL and OWL Full. One of the motivations of having the iff-semantics in the RDF-compatible semantic for OWL is to solve the 'too few entailment' problem [7]. Note that the iff-semantics is *not* relevant to the direct semantics of OWL DL.

Among the 15 OWL URIs, the pD* interprets `owl:FunctionalProperty`, `owl:InverseFunctionalProperty`, `owl:SymmetricProperty` and `owl:TransitiveProperty` as the if conditions of the standard mathematical definitions. The `owl:inverseOf` is interpreted as that if two properties are owl:inverseOf-related, then their extensions are each other's inverse as binary relations. The pD* semantics requires that two classes are equivalent if and only if they are both subclasses of each other. `owl:equivelantProperty` is treated in a similar way to `owl:equivalentClass`. The pD* semantics interprets `owl:sameAs` as an equivalence relation.In particular, the pD* semantics includes the iff condition for `owl:hasValue`. But for `owl:someValueFrom` and `owl:allValueFrom`, the pD* semantics still includes half of OWL's iff conditions. If two classes are owl:disjointWith-related the pD* semantics requires their extensions are disjoint. The pD* semantics requires that the extensions of `owl:sameAs` and `owl:differentForm` are disjoint. Given the pD* semantics discussed above, the corresponding pD* entailment rules are also given in [6]. It consists of 23 rules to illustrate that what conclusion can be deduced from some given premises. These rules are proved to be sound and complete with respect to the pD* semantics.

With respect to the subset of the OWL vocabulary considered, the pD* semantics is intended to represent a reasonable interpretation that is useful for drawing conclusions about instances in the presence of an ontology, and that leads to simple entailment rules and a relatively low computational complexity.

4 A Forward-Chaining Reasoning Algorithm

4.1 Sesame Algorithm

Sesame is a Java framework for storage and querying of RDF and RDFS information. It uses a forward-chaining algorithm to compute and store the closure during any transaction that adds data to the repository. The algorithm runs iteratively over the RDFS entailment rules, but makes use of the dependencies between entailment rules to eliminate redundant inferring steps. Where, a $rule(r1)\ a1 \rightarrow b1$ triggers another $rule(r2)\ a2 \rightarrow b2$ if there is some conclusion $s \in b1$ that matches a premise $p \in a2$. Such a trigger is referred to a dependency between two rules. The dependency relations between the entailment rules used in sesame algorithm are shown in the left of Table1.

The Sesame algorithm is guaranteed to terminate: each new iteration is applied only to statements newly derived in the previous iteration. Since the total set of statements in the closure is finite, the algorithm terminates when no new statements can be derived.

4.2 Optimized Dependencies Between RDFS Entailment Rules

Based on the Sesame algorithm, we provide an optimization to the dependencies between RDFS entailment rules for a given premise. Using this optimization, much redundant inferring steps can be eliminated when computing the pD* closure. Usually, a knowledge base is divided into two levels: schema level and instance level. Schema level contains the statements about concepts and the relationships between concepts. Instance level contains the statements about individuals. Besides these two levels, we define a metaschema level. This metaschema level contains the declarations about built-in vocabulary. We define that the metaschema level contains the statements which satisfy such pattern: the subject is the built-in vocabulary, the predict is `rdf:type` or `rdfs:subClassOf` or `rdfs:subPropertyOf` or `rdfs:domain` or `rdfs:range` or `owl:sameAs` or `owl:equi- valentClass` or `owl:equivalentProperty`. That is to say, the metaschema level contains the statements such that what is the domain of `rdf:type` or what is the range of `rdf:type`. Obviously, the axiomatic triples and the statements deduced from them are included in the metaschema level.

Usually, there are no metaschema level statements in an usual RDF or OWL file. That is to say, user will hardly make the statements like what is the domain of `rdf:type`. They always accept the axiomatic triples as default. Based on this hypothesis, the optimized result to the dependent table is shown in the right of Table1. Take the dependency between rule1 and rule3 for example, the conclusion of rule1 is: <p type Property>. The premises of rule3 are: <p range

Table 1. The dependent and optimized dependent table

	1	2	3	4a	4b	5	6	7	8	9	10	11	12	13		1	2	3	4a	4b	5	6	7	8	9	10	11	12	13
1		●	●	●			●	●		●					1		●		●			●							
2		●	●				●	●	●	●	●		●	●	2			●				●		●	●	●		●	●
3		●	●				●	●	●	●	●		●	●	3			●				●		●	●	●		●	●
4a		●	●					●		●					4a														
4b		●	●					●		●					4b														
5						●		●							5						●		●						
6		●	●					●							6														
7		●	●			●	●	●	●	●	●	●	●	●	7		●	●			●	●	●	●	●	●	●	●	●
8		●	●					●				●			8												●		
9			●				●	●	●	●	●		●	●	9			●				●		●	●	●		●	●
10		●	●					●							10														
11								●		●		●			11										●		●		
12		●	●		●	●		●							12						●		●						
13		●	●					●		●		●			13										●		●		

u>, <v p w>. We may find that the conclusion of rule1 matches the second premise of rule3, because there has the triple: <type range Class> in RDFS axiomatic triples and we suppose that user accepts it as default. When rule1 triggers rule3, only statement: <Property type Class> which has been already deduced from axiomatic triples can be inferred. So the dependency between rule1 and rule3 is meaningless, it can be removed. Compared to the dependent table used in sesame algorithm we can find that 40% of dependent relationships are removed using this optimization.

4.3 The Reasoning Algorithm

Applying the optimized method to all the entailment rules (RDFS entailment rules and pD* entailment rules), a complete dependent table between all entailment rules can be illustrated. We do not list the complete dependent table for the reason of limited space.

Using the complete dependent tables, we propose a forward-chaining reasoning algorithm. It consists of a simple loop to obtain the pD* closure of an RDF graph G. It is an iterative procedure to apply the entailment rules to G and terminate until no new statements can be derived. The detailed algorithm is described as follows:

1. Initialize all rules are recorded as triggered.
2. Read in RDF graph G and all the axiomatic triples.
3. Begin iteration.
4. For each rule, determine whether it is triggered in last iteration. If its premises are matched by newly derived triples in last iteration, apply this rule to graph G and record the rules triggered by it.
5. Iteration terminate until no new triple added to G.

When a certain rule is applied to G, it means that firstly we search the triple newly derived in last iteration for the triple matches one of the premises of the rule, if succeed, try to find the other premises in G, if all premises can be matched, the conclusion of this rule is deduced and added to G. For example, when rule rdfs2 is triggered, we will firstly search all the triples derived in last iteration to find a triple that matches at least one of the premises. If succeed, then search G for the triples matched the other premise. A pair of triples that matched with these two premises will deduce the conclusion triple, all such pairs are found out and the corresponding conclusions are derived. Finally, the rules triggered by rdfs2 are recorded for next iteration.

Using this algorithm, the pD* closure(Gp) of G can be computed in polynomial time. And whether G pD* entails RDF graph H can be converted into checking if Gp contains an instance of H as a subset or contains a P-clash. A P-clash is either a combination of two triples of the form <v differentFrom w>, <v sameAs w>, or a combination of three triples of the form <v disjointWith w>, <u type v>, <u type w>. Same as Sesame algorithm, this algorithm is also guaranteed to terminate.

5 Test Results

The OWL Web Ontology Language Test Cases [2] is a W3C Recommendation. Because there does not have the real pD* test cases, we test our algorithm on the positive entailment test cases of OWL. We select all the test cases responding to the vocabulary supported by pD* from [2]. The results are shown in Table2. For each OWL test case, the symbol '—' indicates that it is not a positive entailment test case, otherwise, there are two denotations. The symbol "S" (or "U") at the left position indicates that the underlying semantic condition of this test case is supported totally(or unsupported) by pD*, while the symbol "P" ("U"or "F") at the right position indicates that our algorithm passes (unsupports or fails in) the corresponding test case.

Totally, there are 37 positive entailment tests about the OWL vocabulary subset included by pD*. From the test results we observe that the underlying semantic conditions of 18 test cases are supported totally by pD*. Among them, our algorithm passed 16 tests. Take the first test case of `owl:allValuesFrom` for instance, our algorithm will apply firstly the entailment rule rdfs9 to infer from <i type r> and <r subClassOf _:a> that <i type _:a> is tenable. Then after applying entailment rule rdfp16, we infer from <i type _:a>, <_:a onProperty p>, <_:a allValuesFrom c>, <i p o> that <o type c> is tenable. Since <o type Thing> and <c type Class> are tenable in the premises, the conclusions of this test case are all tenable. So this test case is passed. The other passed test cases are similar to this example. Among the 18 pD* test cases, two of them which includes datatype are failed because our algorithm does not support the reasoning of datatype by now. The test results listed above illuminate that with respect to the pD* most of the test cases can be passed by our algorithm.

Table 2. Test results

Positive Entailment Test	001	002	003	004	005	006	007
FunctionalProperty	S/P	S/F	U/U	U/U	U/U	—	—
InverseFunctionalProperty	S/P	S/F	U/U	U/U	—	—	—
Restriction	—	—	—	—	—	U/U	—
SymmerticProperty	S/P	U/U	U/U	—	—	—	—
TransitiveProperty	S/P	U/U	—	—	—	—	—
allValuesFrom	S/P	—	—	—	—	—	—
differentFrom	U/U	U/U	—	—	—	—	—
disjointWith	S/P	S/P	—	—	—	—	—
equivalentClass	S/P	S/P	S/P	U/U	—	U/U	U/U
equivalentProperty	S/P	S/P	S/P	U/U	U/U	S/P	—
inverseOf	S/P	—	—	—	—	—	—
sameAs	S/P	—	—	—	—	—	—
someValuesFrom	U/U	U/U	U/U	—	—	—	—

For illustrating the performance of our algorithm, we use five different data sets to test the loading time of three different algorithms. One of them is the exhausitive forward-chaining algorithm which does not use the dependencies between rules, one is our algorithm discussed above, the other is a more simple algorithm in which some entailment rules are taken off for promoting the performance. The reasoning results of this simple algorithm are guaranteed to be sound, but may be incomplete. The test results show that the data loading time of our algorithm using optimized dependencies between rules is better than exhausitive forward-chaining algorithm. Among these three algorithms, the performance of simple algorithm is the best.

From the test results listed above, we find that with respect to the pD* most of the test cases can be passed by our algorithm and its data loading time is better than exhausitive forward-chaining algorithm. In addition for the users who want to get some usual results rapidly but does not like to wait for complete reasoning results, the simple algorithm is more suitable.

6 Conclusion and Future Work

In this paper, we have presented a forward-chaining reasoning algorithm which supports the reasoning of pD*. Based on the premise that metaschema level statements are usually absent in users' RDF or OWL files, an optimization to the dependencies between entailment rules is applied for elevating the algorithm's performance. The test results shows that its data loading time is better than exhaustive forward-chaining algorithm. In addition, we also provide an efficient approximate algorithm for users who want some correct results rapidly but do not require the quick answers to be complete.

The work reported in this paper can be seen as a first step towards a complete system for storing and querying Semantic Web data with pD* semantics. There are a lot of works to do towards this direction, such as to deal with the consequences of delete operations, to improve the performance for scalability. How to solve these problems will be discussed in our future work.

Acknowledgments

The work is supported partially by the 973 Program of China under Grant 2003CB317004, the JSNSF under Grant BK2003001 and the European project Knowledge Web FP6 Network of Excellence EU project Knowledge Web (IST-2004-507842). We would like to thank our team members for their work on related experiments.

References

1. Broekstra, J., Kampman, A., Harmelen, F.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In Proc.of the first International Semantic Web Conference (2002), pp. 54-68.
2. Carroll, J.J., Roo, J.D. (Eds.): OWL Web Ontology Language Test Cases. W3C Recommendation 10 February 2004. http://www.w3.org/TR/2004/REC-owl-test-20040210/.
3. Guo, Y., Pan, Zh., Heflin, J.: An Evaluation of Knowledge Base Systems for Large OWL Datasets. In Proc. of the 3rd International Semantic Web Conference (2004), pp. 274-288.
4. Haarslev, V., Moller, R.: RACER system description. In Proc. of the Int. Joint Conference on Automated Reasoning (IJCAR 2001), volume 2083 of Lecture Notes in Artificial Intelligence, pp. 701-705.
5. Hayes, P. (Ed.): RDF Semantics. W3C Recommendation 10 February 2004. Latest version is available at http://www.w3.org/TR/rdf-mt/.
6. Horst, H.J.: Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. Journal of Web Semantics 3 (2005), pp. 79-115.
7. Pan, J.Z.,Horrocks, I.: RDFS(FA) and RDF MT: Two Semantics for RDFS. In Proc. of the 2nd International Semantic Web Conference (ISWC2003), pp. 30-46.
8. Patel-Schneider P. F.: Building the Semantic Web Tower from RDF Straw. In Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005).
9. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur A., Katz, Y.: Pellet: A Practical OWL-DL Reasoner. Submitted for publication to Journal of Web Semantics.
10. Tsarkov, D., Horrocks, I.: Efficient reasoning with range and domain constraints. In Proc. of the Description Logic Workshop (2004), pp. 41C50.

Triple Space Computing: Adding Semantics to Space-Based Computing

Johannes Riemer[1], Francisco Martin-Recuerda[2], Ying Ding[3], Martin Murth[1],
Brahmananda Sapkota[4], Reto Krummenacher[2], Omair Shafiq[2],
Dieter Fensel[3], and Eva Kühn[1]

[1] Institute of Computer Languages, Space-Based Computing Group, Vienna University of Technology, Vienna, Austria
{mm, jr, eva}@complang.tuwien.ac.at
[2] Digital Enterprise Research Institute, University of Innsbruck, Innsbruck, Austria
{francisco.martin-recuerda, reto.krummenacher, omair.shafiq}@deri.org
[3] Electronic WebService GmbH, Innsbruck, Austria
[4] Digital Enterprise Research Institute, National University of Ireland, Galway, Galway, Ireland
{brahmananda.sapkota}@deri.org

Abstract. Triple Space Computing (TSC) is a very simple and powerful paradigm that inherits the communication model from Tuple Space Computing and projects it in the context of the Semantic Web. In this paper, we propose Triple Space Computing as a new communication and coordination framework for Semantic Web and Semantic Web Services. We identify the value added by TSC and propose the overall architecture of TSC and the interactions among different components.

1 Introduction

Triple Space Computing [1] is a powerful paradigm that inherits the communication model from Tuple Space Computing and projects it in the context of the Semantic Web. Instead of sending messages forward and backward among participants, like most of today's web service-based applications do, triple-based applications just use a simple communication based on reading and writing RDF triples [2] in shared persistent and semantically described information spaces. Triple Space Computing as a new paradigm for coordination and communication compliant with the design principles of the Web, thus provides a major building block for the Semantic Web and for interoperation of Semantic Web Services.

The current communication paradigm of Web Services is message-oriented. SOAP as a communication technology for XML implies messaging, WSDL defines messages that a Web Service exchanges with its user, and literally all ongoing research efforts around Semantic Web Services rely on these technologies. Triple Space Computing (TSC) [1] aims to promote a promising alternative to message-based communication technologies by adding semantics to Tuple Space computing [3]. TSC is based on the evolution and integration of several well-known technologies: Tuple Space Computing, Shared Object Space [4], Semantic Web and in particular RDF [2]. Tuple Space

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 300 – 306, 2006.
© Springer-Verlag Berlin Heidelberg 2006

computing was invented by David Gelernter in the mid-80s at Yale University. Initially presented as a partial language design, Linda was then recognized as a novel communication model on its own and is now referred to as a *coordination language* for parallel and distributed programming. Coordination provides the infrastructure for establishing communication and synchronization between activities and for spawning new activities. There are many instantiations or implementations of the Linda model, embedding Linda in a concrete host language. Examples include C-Linda, Fortran-Linda and Shared-Prolog. Linda allows defining executions of activities or processes orthogonal to the computation language, i.e. Linda does not care about, how processes do the computation, but only *how* these processes are created. The Linda model is a *memory* model. The Linda memory is called *Tuple Space* and consists of logical tuples. There are two kinds of tuples. Data tuples are passive and contain static data. Process tuples or "live tuples" are active and represent processes under execution. Processes exchange data by writing and reading data tuples to and from the Tuple Space.

However, [1] reports some shortcomings of the current Tuple Space models. They lack any means of name spaces, semantics, unique identifiers and structure in describing the information content of the tuples. TSC takes the communication model of Tuple Space Computing, wherein communication partners write the information to be interchanged into a common space and thus do not have to send messages between each other; TSC enhances this with the semantics required for Semantic Web enabled technologies.

The prototype development is based on Corso (Coordinated Shared Objects) system [4]. Corso is a platform for the coordination of distributed applications in heterogeneous IT environments that realizes a data space for shared objects. Corso offers maximum scalability and flexibility by allowing applications to communicate with one another via common distributed persistent „spaces of objects“. For testing and validating the TSC technology with special attention to the support for Semantic Web Services, we will integrate this system in the Semantic Web Service Environment WSMX[1], which is the reference implementation of the Web Service Modeling Ontology WSMO[2]. Thereby, the TSC technology will be aligned with emerging technologies for Semantic Web Services. By providing the basis for a new communication technology for the Semantic Web, TSC will provide a significant contribution to international research and development efforts around the Semantic Web and Semantic Web Services.

In this paper, we report some of the progresses. We present the overall architecture of TSC, which is mainly focusing on the TSC data and operation model and the introduction of different components involved in a Triple Space environment and the connections and interaction among these components. Finally we mention some potential future works.

2 TSC Architecture

Like the Web, TSC originally intended to build a Triple Space Computing infrastructure based on the abstract model called REST (Representational State Transfer) [6].

[1] www.wsmx.org

[2] www.wsmo.org

The fundamental principle of REST is that resources are stateless and identified by URIs. HTTP is the protocol used to access the resources and provides a minimal set of operations enough to model any application domain [6]. Those operations (GET, DELETE, POST and PUT) are quite similar to Tuple Space operations (RD, IN and OUT in Linda)). Tuples can be identified by URIs and/or can be modeled using RDF triples. Since every representation transfer must be initiated by the client, and every response must be generated as soon as possible (the statelessness requirement) there is no way for a server to transmit any information to a client asynchronously in REST. TSC project members are evaluating several extensions of REST, like ARRESTED [7] that can provide a proper support of decentralized and distributed asynchronous event-based Web systems.

TSC envision a decentralized and distributed infrastructure of nodes that provide all the services that Triple Space Computing requires. TSC promotes a hybrid architecture that combines the advantages of pure P2P and client/server systems, called super-peer systems [8]. This configuration drives into a two-tiered system. The upper-tier is composed of well-connected and powerful servers, and the lower-tier, in contrast, consists of clients with limited computational resources that are temporarily available. Three kinds of nodes are identified in Triple Space architecture:

- **Servers.** store primary and secondary replicas of the data published; support versioning services; provide an access point for light clients to the peer network; maintain and execute searching services for evaluating complex queries; implement subscription mechanisms related with the contents stored; provide security and trust services; balance workload and monitor requests from other nodes and subscriptions and advertisements from publishers and consumers.
- **Heavy-clients.** are peers that are not always connected to the system. They provide most of the infrastructure of a server (storage and searching capabilities) and support users and applications to work off-line with their own replica of part of the Triple Space. Replication mechanisms are in charge to keep replicas in clients and servers up-to date.
- **Light-clients.** only include the presentation infrastructure to write query-edit operations and visualize data stored on Triple Spaces.

Servers and heavy-clients are in charge of running Triple Space kernels (TS kernels). A TS kernel provides interfaces for participants to access Triple Spaces and implements the persistent storage of named graphs [9], synchronization of participants via transactions and access control to Triple Spaces. A Triple Space can be spanned by one or multiple TS kernels. If multiple TS kernels are involved, the kernels exchange the named graphs of the space in a consistent way. Participants are users and applications which use the Triple Space in order to publish and access information and to communicate with other participants. Applications can be run in servers, heavy clients, and light clients.

2.1 Triple Space Data Model and Operations Summary

A Triple Space contains data in form of non-overlapping named graphs. A named graph consists of a name, which is a URI, and an RDF graph. Participants read, write and take named graphs to and from the Triple Space in the same way as tuples are read, written and taken in Tuple Spaces. The essential difference is that named graphs can be related to each other via the contained RDF triples. For example the object of a triple in one named graph can be the subject of a triple in another named graph. This way named graphs are not self-contained (as tuples in Tuple Spaces), but can build arbitrary RDF graphs to represent information. To make use of such nested triples, the TSC interaction model provides, in addition to the already mentioned operations, a query operation. This operation allows creating new RDF graphs out of the named graphs in a given space. Mediation based on RDF Schema allows overcoming heterogeneity of data. All Triple Space operations are performed against a certain Triple Space, which is identified by a Triple Space URI.

The TSpace API[3] has been adapted for reading (take, waitToTake, read, waitToRead and scan) and publishing (write) tuples in Triple Space [10]:

- `write (URI space, Transaction tx, Graph g): URI`
  Write one RDF graph in a concrete Triple Space and return the URI of the created named graph. The operation can be included as a part of a transaction.
- `query (URI space, Transaction tx, Template t): Graph`
  Execute a query over a Triple Space identified by an URI. The operation can be included as a part of a transaction.
- `waitToQuery (URI space, Transaction tx, Template t, TimeOut timeOut): Graph`
  Execute a query over a Triple Space identified by an URI. It is similar to the query operation but waits until a given time to return an RDF graph. This is a blocking operation and supports transaction.
- `take (URI space, Transaction tx, URI namedGraphURI|Template t): NamedGraph`.
  Return the named graph identified by URI, respectively a named graph (or nothing) that matches with the template, specified as a parameter. The named graph is deleted, and the operation can be included as a part of a transaction.
- `waitToTake (URI space, Transaction tx, Template t, Timeout timeOut): NamedGraph`
  Like `take` but the process is blocked until the a name graph is retrieved.
- `read (URI space, Transaction tx, URI namedGraphURI| Template t): NamedGraph`.
  Like `take` but the named graph is not removed.
- `waitToRead (URI space, Transaction tx, Template t, Timeout timeOut): NamedGraph`.
  Like `read` but the process is blocked until a named graph is retrieved.
- `update(URI space, Transaction tx, NamedGraph ng): boolean`
  `This` operation allows to update a given named graph in the space. This operation supports transactions. The internal semantics of the operation is `take` and `write` in that order.

[3] http://www.almaden.ibm.com/cs/TSpaces/Version3/ClientProgrGuide.html

The inability of Tuple Space computing to provide *flow decoupling* from the client side [11] is solved by extending the Tuple Space computing model with subscription operations [10]. Thus, two main roles for participants are defined: **producers**, which publish information and advertisements (description of which information will be published); and **consumers**, which expresses its interest in concrete information by publishing subscriptions. The new extensions based on SIENA API[4] [12] can be found in [10].

Finally, transaction support is included to guarantee the successful execution of a group of operations (or the abortion of all of them if one fails). Transactions have been proposed in several Tuple Space computing implementations like TSpaces and JavaSpaces[5]. The new extensions for transaction support can be found in [10].

2.2 Triple Space Kernel Overview

The TS kernel is a software component which can be used to implement both Triple Space servers and heavy clients. In the former case it also provides a proxy component, which allows light clients to remotely access the server. The TS kernel itself consists of multiple components shown in Fig. 1.

TS operations and security layer accepts Triple Space operations issued by participants via the *TSC API. Heavy clients* run in the same address space as the TS kernel, and the TS kernel is accessed by its native interface. *Light clients* use *TS proxies* to access the TS kernel of a server node transparently over the network. As a variation a light client can access a TS kernel via a standardized protocol, e.g. HTTP. In this case a server side component, e.g. a *servlet*, translates the protocol to the native TS kernel interface. The execution of a TS operation includes verification of security constraints, maintaining state of blocking operations and invocation of the underlying coordination layer. The *security management API* is used to define and change security configurations such as access control for spaces or named graphs. The *coordination layer* implements transaction management and guarantees that concurrent operations are processed consistently. It accesses the local *data access layer* to retrieve data from a space and to apply permanent changes to a space. Furthermore, if a space is spanned by multiple TS kernels, the coordination layer is responsible for inter-kernel communication to distribute and collect data and to assure that all involved kernels have a consistent view to a space. The *mediation engine* resolves heterogeneity issues by providing mappings for possibly occurring mismatches among different RDF triples. It is due to the possibility that different participants may have different RDF schemas while communicating via Triple Space. Mapping rules for mediation are provided to the mediation engine at design time and are processed during run time in order to resolve heterogeneities by identifying mappings. The *mediation management API* provides methods to turn on/off the usage of the mediation engine and to add, remove and replace mediation rules.

The coordination layer is based on the middleware Corso, which is used to replicate named graphs to all involved TS kernels and guarantees consistency via built-in transactions. YARS [13] is used to realize the data access layer.

[4] http://www-serl.cs.colorado.edu/serl/siena/

[5] http://java.sun.com/developer/products/jini/index.jsp

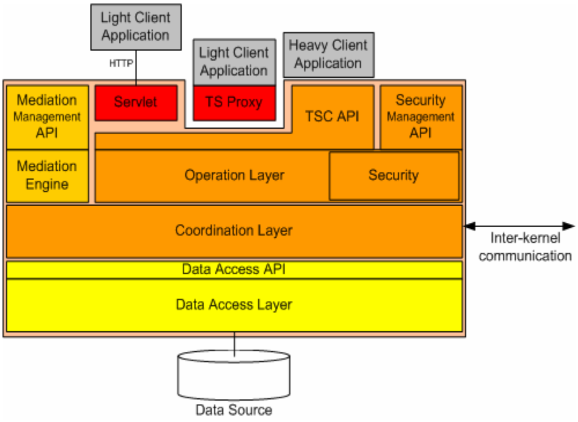


Fig. 1. The TS kernel

3 Conclusion and Future Work

In this paper, we describe the current status of the development of Triple Space Computing as a novel communication and coordination framework that combines Semantic Web technologies and Tuple Space computing for Semantic Web Services. We have conducted current state of the art studies in related fields and identify the value added by TSC. Based on this, we propose the overall architecture of TSC and explain the interactions among different components.

TSC is an Austrian national funded project which still has two years to go. During these two years, we will provide a consolidated TSC architecture and interfaces for cooperation among the components and for the TSC infrastructure as a whole, especially design mediation and query engine components for TSC. Furthermore, we will focus on data replication, security and privacy mechanisms in TSC, to investigate the relation between WSMO[6] and the TSC conceptual model and to find out how standard architectures (REST, SOA) can be better applied in TSC. In the end, a running prototype will be provided and the usability will be tested via a case study on how TSC can enhance communication and process coordination in WSMX[7].

Acknowledgement

The work was supported by the Austrian Federal Ministry for Transport, Innovation, and Technology under the FIT-IT Semantic Systems project Triple Space Computing (TSC 809249). The authors would like to thank the people who are involved in TSC project and the funding support from the Austrian government.

[6] http://www.wsmo.org
[7] http://www.wsmx.org

References

1. D. Fensel: Triple Space computing: Semantic Web Services based on persistent publication of information, the IFIP Int'l Conf. on Intelligence in Communication Systems, 2004.
2. Klyne, G. and Carroll, J. J. (eds.): Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, February 2004, available at http://www.w3.org/TR/rdf-concepts/
3. Gelernter, D.: Generative Communication in Linda. ACM Trans. Prog. Lang. and Sys. 7(1) (1985) 80-112
4. Kühn, E.: Fault-Tolerance for Communicating Multidatabase Transactions. Proc. Of the 27th Hawaii Int. Conf. on System Sciences (HICSS), ACM, IEEE (1994)
5. Johanson, B., Fox, A.: Extending Tuplespaces for Coordination in Interactive Workspaces. J. Systems and Software, 69(2004) 243-266.
6. Fielding, R. T.: Architectural Styles And The Design of Network-based Sofware Architectures. PhD Thesis, University of California, Irvine (2000)
7. Khare, R., Taylor, R. N.: Extending the Representational State Transfer Architectural Style For Decentralized Systems. Proc. Of The International Conference on Software Engineering (ICWE), Edinburgh, Scotland (2004)
8. Yang, B., Garcia-Molina, H.: Designing a Super-Peer Network. IEEE International Conference On Data Engineering (2003)
9. Carroll, J. J., Bizer Ch., Hayes, P., Stickler, P.: Named Graphs. J. of Web Semantics 3 (2005)
10. Riemer, J., Martin-Recuerda, F., Murth, M., Sapkota, B., Shafiq, O. D2.1 v1.0 TSC Overall Architecture and Components Integration March 15, 2006. http://tsc.deri.at/deliverables/D21.html
11. Eugster, P. T., Felber, P.A., Guerraoui, R., Kermarrec, A. M.: The Many Faces of Publish/Subscribe. ACM Computing Survey (2003)
12. Carzaniga, A.: Architectures For An Event Notification Service Scalable to Wide-Area Networks. PhD Thesis. Politecnico di Milano. (1998)
13. Harth, A.: Optimized Index Structures For Querying RDF From The Web (2005)

Full-Automatic High-Level Concept Extraction from Images Using Ontologies and Semantic Inference Rules*

Kyung-Wook Park and Dong-Ho Lee**

Department of Computer Science and Engineering, Hanyang University,
Ansan-si, Gyeongki-do 426-791, South Korea
{kwpark, dhlee72}@cse.hanyang.ac.kr

Abstract. One of the big issues facing current content-based image retrieval is how to automatically extract the semantic information from images. In this paper, we propose an efficient method that automatically extracts the semantic information from images by using ontologies and the semantic inference rules. In our method, MPEG-7 visual descriptors are used to extract the visual features of image which are mapped to the semi-concept values. We also introduce the visual and animal ontology which are built to bridge the semantic gap. The visual ontology facilitates the mapping between visual features and semi-concept values, and allows the definition of relationships between the classes describing the visual features. The animal ontology representing the animal taxonomy can be exploited to identify the object in an image. We also propose the semantic inference rules that can be used to automatically extract high-level concepts from images by applying them to the visual and animal ontology. Finally, we discuss the limitations of the proposed method and the future work.

1 Introduction

Recently, the explosive growth of the Internet and mobile device has established the need for the development of tools for the more efficient retrieval of multimedia contents. However, traditional retrieval approaches such as keyword-based or content-based image retrieval have a number of limitations for a large image database. In the case of keyword-based approach using the predefined vocabulary, image contents have to be manually annotated by domain experts. Therefore, this method needs the large amount of manual effort required in developing the annotations of image collections, and suffers from the inconsistency of the keyword assignments among different indexers.

To overcome the limitations of the keyword-based approach, various content-based image retrieval (hereinafter, called CBIR) methods [1,2,3] have been proposed in recent years. Most of CBIR systems provide more user-friendly access facilities on images by their content through queries such as "*Retrieve all images that are similar to*

[*] This work was supported by Korea Research Foundation Grant funded by Korea Government(MOEHRD, Basic Research Promotion Fund, KRF-2005-003-D00288).

[**] Corresponding author.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 307 – 321, 2006.
© Springer-Verlag Berlin Heidelberg 2006

this image in its color", where *this image* refers to a hand-drawn sketch or an example image provided by users and *the color* refers to its content. In such a system, the visual features such as color, texture, and shape of the image are (semi-)automatically extracted and stored. A query image's content is extracted during run-time and used to match against those in large image database. Although these systems are less time-consuming and provide more user-friendly access on images than the keyword-based approach, they still have several problems such as follows: First, the visual features can not be interpreted to high-level concepts that human can understand more easily. In other words, a human can not directly understand the visual features because they are commonly represented by high-dimensional feature vectors. Also, the visual similarity does not necessarily mean the semantic similarity. For example, if user requests an image with a 'red rose' to a CBIR system, it is likely to answer images with a 'red ball'. That is, there is a gap between the visual features of an image and semantic information. This problem is called 'Semantic Gap' by researchers of this field. Therefore, it is difficult for most CBIR systems to correctly respond to the semantic queries such as 'Lion on the ground' or 'Lion at the zoo' if the semantic gap is not solved. Second, even if an image contains the homogeneous objects, it may have different visual features since the visual features may be influenced by the physical environments like the light and the camera angle. The fundamental aim of the image retrieval is to make users more efficiently and easily find the images that they want to find from large image database. However, as mentioned above, it is hard to achieve this aim by traditional CBIR methods without any semantic information (i.e., high-level concept).

To overcome such drawbacks, a few researches have been done on using ontologies for retrieval of visual resources such as image and video [4,5,6,7]. Ontology is a type of background knowledge that defines all of the important categories of concepts that exist in a specific domain, and the relationships between them. In early researches on image retrieval using ontology, ontologies are just used for assisting manual annotation [4,5]. These ontologies are not suitable for automatic annotation since they contain little visual information about the concepts they describe. Therefore, a few researches now are investigating how to automatically annotate the high-level concepts in the image by using ontologies [6,7]. They employed ontologies to automatically annotate the high-level concepts from the visual features which are extracted by image processing techniques. However, the ontologies still have problems since they employed too simple ontologies.

In this paper, we propose a novel method, which links the visual features to semi-concept values and then automatically extract the high-level concept which exist in an image by applying the inference rules to the visual and animal ontology. In our method, the visual ontology facilitates the mapping between the visual features and the semi-concept values and allows the definition of relationships between the classes describing the visual features. Our visual ontology is a generic ontology that is not confined to the specific domain ontology (here, the animal ontology). Namely, it is possible to apply the visual ontology for various domains, such as the cell images and other image domains. Also, the animal ontology representing animal taxonomy is used to identify high-level concepts in the inference phase. In order to identify the object from the visual and animal ontology, we also define the inference rules which are applied to them. In the proposed system, the evaluation domain, animal images, is finally presented – together with the high-level concept and additional information such as the creation time, comment, and image size.

The rest of this paper is organized as follows: In section 2, we briefly review related work. Section 3 describes the structure of the proposed system and the method that extracts the high-level concept existing in an image from the visual features. Section 4 introduces the design and implementation of the proposed system. Finally, we conclude our work and discuss our future plans in Section 5.

2 Related Work

A number of research efforts have investigated the pattern recognition techniques which include statistically-based, machine-learning methods to extract visual features which can be used to generate semantic descriptions of multimedia content [4,5,6,7,8]. Most of these researches first manually annotate sample sets, and then generate statistical models for the larger data collection. However, these researches usually suffer from the difficulty of the design and configuration of multiple variables and options.

Jianping Fana *et al* [8] proposed a multi-level framework consisting of two steps for bridging the semantic gap, where the concept-sensitive salient objects are used to enable more expressive representation of image contents. However, in the proposed algorithm, the training corpora are required to ensure optimum performance, but these cannot easily be adapted to new domains or incorporate knowledge. In addition, users must find the best parameters for each concept-sensitive salient object.

To overcome these drawbacks, various image retrieval approaches using ontology have been proposed by several researchers in recent years. As mentioned above, ontology helps to extract semantic concepts form images and facilitates retrieval in a more convenient way. In [6], the authors proposed ontology on four types of artworks. The proposed ontology includes various kinds of concepts that allow users query the visual features in various aspects. However, the proposed method basically depends on object detection techniques. Vasileios Mezaris *et al* [7] proposed the method which uses an object ontology and the intermediate-level descriptor values to describe semantic information. The object ontology, which has each object as the top-level concept, is a specification for a specific domain object and includes the human-readable intermediate-level descriptor values. For example, ‘tiger’ object is defined as Luminance = {high, medium}, green-red = {red low, red medium}, blue-yellow = {yellow medium, yellow high} and size = {small, medium}, where the intermediate-level descriptor values such as ‘high’ and ‘red low’ are defined based on the visual features of a specific domain object. However, the proposed method has several drawbacks as follows: First, basic relationship in the object ontology only takes into account the subsumption relationship between an object and the visual feature classes that belong to it. For more detailed description, it is necessary to define the relationship among the objects and the relationship among the visual feature classes as well as the relationship between the object and its visual features. Second, the object ontology must be reorganized every time new objects come from the domain experts. That is, since their approach did not employ any inference rule for extracting the high-level concept, the object ontology must be reorganized as the new object is added. In general, semantic inference rules have been used to derive new knowledge from the knowledge existing in a domain.

In our method, we can infer the high-level concepts that exist in images by defining the inference rules which map particular combinations of visual features to high-level concepts defined in the domain ontology (i.e., the animal ontology). In particular, such rules can be shared and collaboratively modified as the domain understanding changes. Therefore, we only have to redefine the inference rules instead of reorganizing the domain ontology for identifying the new object.

3 System Architecture and High-Level Concept Extraction

In this section, we first introduce the overview of the system architecture and explain the methods which map the visual features to semi-concept values in detail. And then, we describe the method which infers the high-level concept from an image by applying the inference rules to the visual and animal ontology.

Figure 1 depicts the overview of the proposed system. The procedure for extracting the high-level concepts from images through the constructed ontolgies is as follows:

1. The user sets the region of interest (or ROI) for an object which exists in an image.
2. The corresponding object is recognized from the ROI, and then the visual features of the corresponding object are extracted by using the MPEG-7 visual descriptors. In our method, we use the edge histogram descriptor (hereinafter, called EHD), contour-based shape descriptor (hereinafter, called Contour-SD) and color structure descriptor (hereinafter, called CSD) of various MPEG-7 visual descriptors.
3. The extracted visual features are mapped to the semi-concept values and saved as the property values of the subclasses of the *Component* class in the visual ontology. In section 3.1, the mapping methods for each of the visual features will be discussed in detail.
4. In order to extract the high-level concept from the image, we use the inference rules which are applied to the visual and animal ontology. For example, assuming that R_1 is the rule to identify *tiger* object, R_1 can be defined as follows.
   Rule R_1:

$$\begin{array}{l}
\textit{If } vdo: object(?\mathrm{x}) \wedge \\
\quad vdo: hasVD(?\mathrm{x},\ ?\mathrm{y}) \wedge vdo: hasEHD\_Component(?\mathrm{y},\ ?\mathrm{z}) \wedge \\
\quad vdo: S\_EH_0(?\mathrm{z},\ \mathrm{Nondirectional}) \wedge vdo: S\_EH_1(?\mathrm{z},\ \mathrm{45diagonal}) \wedge \\
\qquad\qquad\qquad\qquad\qquad\qquad .... \\
\Rightarrow\ ani: Tiger(?\mathrm{x})
\end{array}$$

R_1 states that object x is the *tiger* object if object x is an object and it has the EHD and its S_EH_0 value is '*Nondirectional*' and S_EH_1 value is '*45diagonal*'. The prefixes *vdo* and *ani* indicate terms from the visual and animal ontology, respectively. By defining various semantic inference rules such as R1 and applying them to the visual and animal ontology, we can identify the object in an image.

5. The high-level concept for the corresponding image, which is represented as OWL-based format, is stored to the image description database with the additional information such as the MPEG-7 visual descriptors.

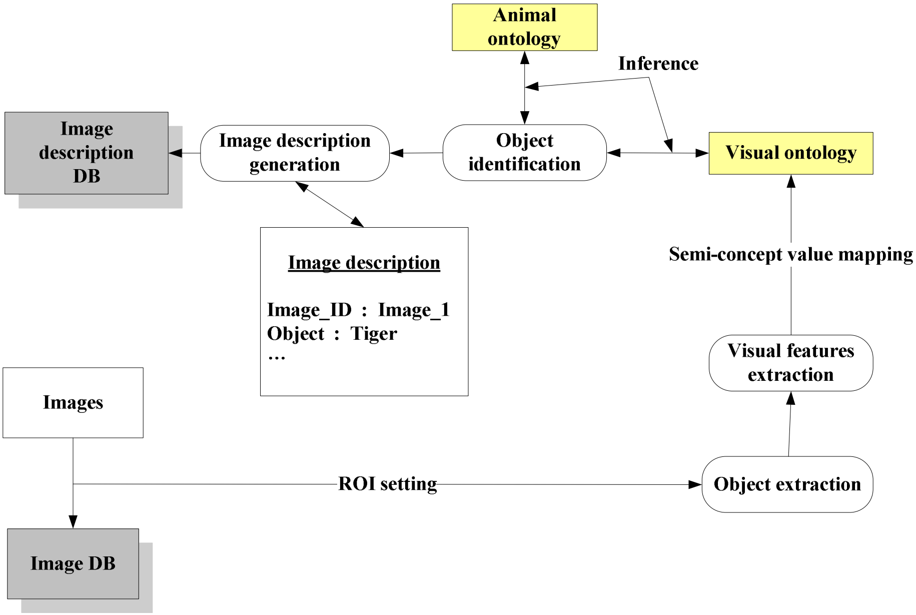


Fig. 1. System architecture

3.1 Semi-concept Value Mapping

MPEG-7 is a standard for describing multimedia content published by the Moving Picture Experts Group (MPEG) [9] and broadly offers a number of tools which describe the multimedia contents in various aspects. In this paper, we employ the MPEG-7 visual descriptors to extract the visual features for the corresponding object. According to Spyrou E. *et al* [10], it is better to exploit several visual features than only one visual feature in the image classification aspect. Therefore, we use the EHD, CSD, and Contour-SD to extract the representative visual features (i.e., texture, color, and shape) which are mapped to the semi-concept values that human can easily understand. Here, the semi-concept values are simple keywords such as ‘low’ or ‘high’ which are automatically assigned by mapping the quantity of the visual features into a specific range.

For the color feature, we make use of the CSD which represents an image by both the color distribution of an image (similar to a color histogram) and the local spatial structure of the color [11]. The color histogram is used to calculate the dominant colors DC_0, DC_1 and the colorfulness $diff_0$, $diff_1$ for the dominant colors DC_0, DC_1. These visual features for the color are mapped into the semi-concept values $S\_DC_{0\sim1}$ and $S\_Diff_{0\sim1}$ such as follows:

1. A 256-bin color histogram is extracted from an input image I represented in the 256 cell-quantized HMMD color space, then bins are unified to a 128-bin color histogram for more efficient computation.

2. The 128-bin color histogram H is defined via five subspaces, i.e. S_m, $m\in\{0,\ldots,4\}$, in the HMMD color space. That is, the color histogram $H = \{S_0,\ldots, S_4\}$, where the subspaces $S_{0\sim2}$ represent the color by dividing *hue* into 8 uniform intervals and *sum* into 4 uniform intervals, giving 8x4 cells in subspaces, respectively. In contrast, S_4 represents the grayscale by dividing *hue* into 1 uniform intervals and *sum* into 16 uniform intervals, giving 16 x 1 cells in Subspace 4. The sub-colors $C_{0\sim7}$ denote the sum of the values (amplitudes) of all the bins belonging to a specific *hue* in the subspaces $S_{0\sim2}$. S_DC_i are defined by

$$C_{j\in\{0,\ldots,7\}} = \sum_{i=0}^{2} S_{ij},$$

$$DC_0 = Max(tot),\ DC_1 = Max(tot - DC_1),$$

$$\text{semi-concept value } S\_DC_{k\in\{0,1\}} = \begin{cases} \text{Red-Orange} & if\ DC_k = C_0 \\ \vdots & \vdots \\ \text{Sea\_Blue-Violet} & if\ DC_k = C_5 \\ \text{Pink} & if\ DC_k = C_6 \\ \text{Red} & if\ DC_k = C_7 \end{cases}$$

where $S_{00} = (h_0 + \ldots + h_3)$, $S_{01} = (h_4 + \ldots + h_7)$, … , $S_{27} = (h_{92} + \ldots + h_{95})$ and the whole *hue* information $tot \in \{C_0, \ldots, C_7\}$.

3. The semi-concept values S_diff_k for the colorfulness $diff_0$, $diff_1$ are calculated as follows:

$$\text{for each } DC_k = C_j, \qquad diff_k = Max(C_j^{S_{ij}}),\ \text{where } C_j^{S_{ij}} = \bigcup_{i=0}^{2} S_{ij}$$

$$\text{semi-concept value } S\_diff_k = \begin{cases} \text{high}, & if\ diff_k = S_{ij},\ \text{where } i = 0 \\ \text{medium}, & if\ diff_k = S_{ij},\ \text{where } i = 1 \\ \text{low}, & if\ diff_k = S_{ij},\ \text{where } i = 2 \end{cases}$$

4. The dominant grayscale DG_2 could also be calculated in a similar way. Note that we do not calculate since the colorfulness of the dominant grayscale DG_2 is zero in the HMMD color space.

In general, an image object can be more easily recognized by the shape feature. MPEG-7 standard offers two kinds of the shape descriptor: region-based shape descriptor and contour-based shape descriptor.

The region-based shape descriptor basically represents pixel distribution within a region. This descriptor is able to describe the complex objects consisting of multiple disconnected regions and relatively faster than the contour-SD. On the other hand, the contour-SD efficiently describes the objects consisting of single contour. In particular, the animal object such as Turtles, Horses and Frogs are much better captured by the contour-SD [11,12]. Therefore, we make use of the contour-SD to describe the shape information because the evaluation domain is the animal images in our work.

The mapping procedure of the ‘GlobalCurvature’ and ‘PrototypeCurvature’ elements of the contour-SD, which represent the eccentricity and circularity values of the image, to the semi-concept values is depicted in Figure 2.

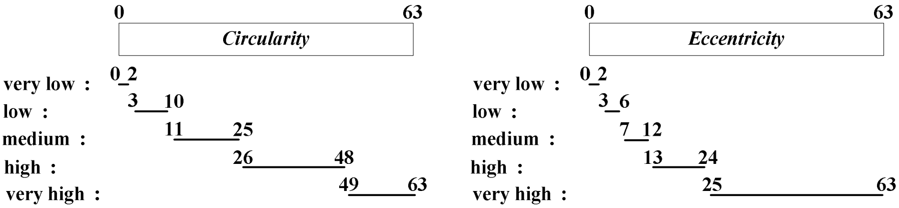


Fig. 2. Correspondence of the two elements of contour-SD and semi-concept values

The EHD represents local edge distribution in an image and is useful for image retrieval, especially for natural images with nonuniform textures [11,13]. An image is first divided into 4 x 4 subimages, then the local-edge distribution for each subimage is represented by a histogram which is calculated for the five types of edges in each subimage (i.e., vertical, horizontal, 45diagonal, 135diagonal and nondirectional). Finally, a total of 5 x 16 = 80 histogram bins are calculated.

In the case of the images containing the homogeneous object, we found the fact that two edge types of the images, which have the maximum bin size in the global-edge histogram, are almost similar. Therefore, we only consider the global-edge histogram *g_EH.* The three edge types E_1^I, E_2^I and E_3^I of the maximum bin values are denoted as the representative edges for an image *I*, where subimage I_{ij} is the subimage of row *i* x column *j* in image *I*. Then, the local-edge histogram $h_{ij} = \{e_0, e_1, e_2, e_3, e_4\}$ and the semi-concept values S_EH_l for the representative edges are as follows.

$$g\_EH = \{E_0, E_1, E_2, E_3, E_4\}, \text{ where } E_{k\in\{0,1,2,3,4\}} = \sum_{i=0}^{3}\sum_{j=0}^{3} e_k^{h_{ij}}$$

$$\begin{cases} E_1^I = Max(g\_EH),\ E_2^I = Max(g\_EH - E_1^I), & \\ E_3^I = Max(g\_EH - (E_1^I \cup E_2^I)), & if(\left|E_2^I - E_3^I\right| \le 2) \\ E_1^I = Max(g\_EH),\ E_2^I = Max(g\_EH - E_1^I), & else \end{cases}$$

$$\text{semi-concept value } S\_EH_{l\in\{1,2,3\}} = \begin{cases} \text{Vertical,} & if\ E_k^I == E_0 \\ \text{Horizontal,} & if\ E_k^I == E_1 \\ \text{45diagonal,} & if\ E_k^I == E_2 \\ \text{135diagonal,} & if\ E_k^I == E_3 \\ \text{Nondirectional,} & if\ E_k^I == E_4 \end{cases}$$

3.2 Building the Visual and Animal Ontology

Ontology is the term referring to the shared understanding of the domain experts in a specific domain, which usually consists of a set of classes (concepts), relationships, and instances. In addition to providing explicit definitions of different concepts and the relationships between them, this makes users more easily retrieval images as they want. The W3C has established the OWL web ontology language [14] on the basis of the Resource Description Framework RDF [15]. We make use of OWL to build the ontologies, since this language facilitates machine interoperability of Web content and offers vocabulary for describing properties and classes.

Figure 3 shows the part of the visual ontology used in our work. It is made up of various classes and relationships between them. Table 1 shows the definition of some classes in the visual ontology. For example, *Object* class, the top-level class, describes the object in an image and is concerned with *VD* class by *hasVD* relationship. In the case of the classes that are not defined in Table 1 (e.g., the *CSD_Component* class), they are similar to the definition of the sibling class.

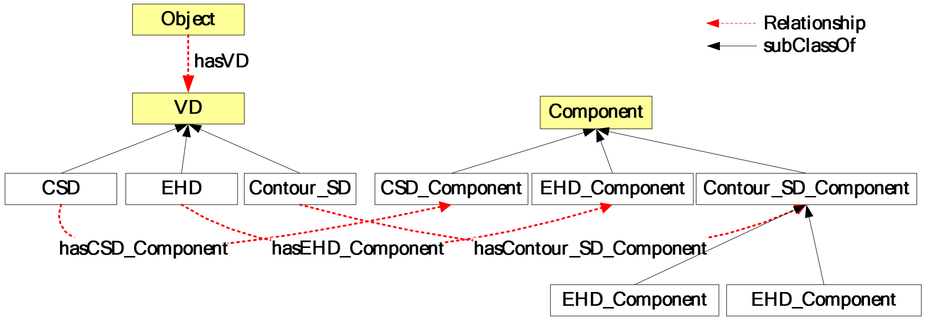


Fig. 3. Class hierarchy and relationships of the classes in the visual ontology

Table 1. The definition of the classes in the visual ontology

Class	Property	Definition
Object	hasVD VD	Describing the object in an image
Component	None	The component of MPEG-7 visual descriptors
VD	None	Describing the MPEG-7 visual descriptors
CSD	∃hasCSD_Component CSD_Component	Describing the CSD of the MPEG-7 visual descriptors
EHD_Component	S_EH0_Value, S_EH1_Value, and S_EH2_Value	The semi-concept values of the EHD

The following OWL document represents the instance of the visual ontology for *Object_1*.

```
<Object rdf:ID = “Object_1”>
   <hasVD rdf:resource = “#EHD_1”/>
   <hasVD rdf:resource = “#CSD_1”/>
   <hasVD rdf:resource = “#Contour_SD_1”/>
</Object>
 <EHD rdf:ID = “EHD_1”>
   <hasEHD_Component>
      <EHD_Component rdf:ID = “EHD_Component_1”>
         <S_EH0_Value rdf:datatype =
           http://www.w3.org/2001/XMLSchema#String>
          Horizontal </S_EH0_Value>
         <S_EH1_Value rdf:datatype =
                 http://www.w3.org/2001/XMLSchema#String>
                45diagonal </S_EH1_Value>
11          </EHD_Component>
12       </hasEHD_Component>
13    </EHD>
14     <CSD rdf:ID = "CSD_1">
15      <hasCSD_Component>
16          <CSD_Component rdf:ID = "CSD_Component_1">
17              <S_DC0_Value rdf:datatype =
                 http://www.w3.org/2001/XMLSchema#String>
                Yellow-Green </S_DC0_Value>
18              <S_DC1_Value rdf:datatype =
                 http://www.w3.org/2001/XMLSchema#String>
                Sky_Blue </S_DC1_Value>
                                 ....
```

In line 1~5, *Object_1* is the instance of the visual ontology and is concerned with *EHD_1*, *CSD_1*, and *Contour_SD_1* instances by *hasVD* relationship. In line 7~13, *EHD_1* instance has the string values '*Horizontal*', '*45diagonal*' as the semi-concept values S_EH_0, S_EH_1. The explanation for the rest is also similar to the above.

Since the evaluation domain is the animal images, we construct the animal ontology representing the animal taxonomy. This ontology is used to infer the high-level concepts by using the inference rules. Figure 4 depicts the class hierarchy of the animal ontology. Although the animal ontology can be expanded continuously, we do not describe all kinds of animal terms because it is very difficult to construct the animal ontology consisting of all kinds of animal terms.

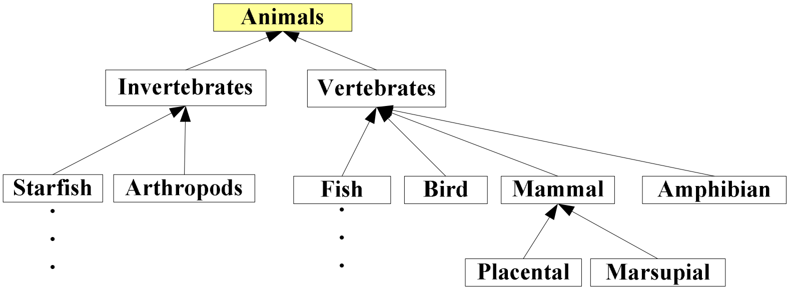


Fig. 4. The class hierarchy of the animal ontology

3.3 Semantic Inference Rules for Extracting the High-Level Concepts

The aim of inference is to derive the new knowledge by applying the inference rules to knowledge existing in a specific domain. Fortunately, various rule engines for OWL reasoning have been proposed [16,17,18]. In this paper, we make use of 'Bossam' rule engine [19] in order to identify an object from the visual and animal ontology. 'Bossam' rule engine basically provides a simple non-markup rule language called 'Buchingae', a rule set for OWL reasoning, and frame-based knowledge representation.

In order to exploit the inference rules for each of the objects, we used the training dataset which consists of about 300 images including eight kinds of animals. As a result of the observation of the semi-concept values for the training dataset, we could only exploit the inference rules for six animal objects with the exception of the some objects which have the irregular semi-concept values. In fact, it is possible to extract different values for the homogeneous object since the visual features of an image are sensitive to the physical environment. In order to illuminate the procedure of extracting the high-level concept from an image by applying the inference rule to the visual and animal ontology, we take as an example object O_1 in Figure 5. The inference procedure is as follows.

1. We assume that the visual ontology for object O_1 has been already generated and the semi-concept values of object O_1 are calculated as follows.

$$
\begin{aligned}
&EHD \in \{S\_EH_0 = \text{"Nondirectional"}, S\_EH_1 = \text{"135diagonal"}\},\\
&Contour\_SD \subset \{Global, Prototype\},\\
&Global \in \{Cir = \text{"very high"}, Ecc = \text{"medium"}\},\\
&Prototype \in \{Cir = \text{"low"}, Ecc = \text{"very low"}\},\\
&CSD \in \{S\_DC_0 = \text{"Red-Orange"}, S\_DC_1 = \text{"Yellow-Green"}, S\_DG = \text{"Black"},\\
&\qquad S\_diff_0 = \text{"Medium"}, S\_diff_1 = \text{"Low"}\}
\end{aligned}
$$

2. *Rule_1* is the rule that is found by the observation of the training dataset including the ‘cheetah’ object. This rule identifies the ‘Cheetah’ object of the animal objects. *Rule_1*:

$$
\begin{aligned}
&\textit{if } vdo:Object(?\text{x}) \wedge\\
&\quad vdo:hasEHD(?\text{x}, ?\text{y}) \wedge vdo:hasEHD\_Component(?\text{y}, ?\text{z}) \wedge\\
&\quad vdo:S\_EH_0\_Value(?\text{z}, \text{"Nondirectional"}) \wedge\\
&\quad vdo:S\_EH_1\_Value(?\text{z}, \text{"135diagonal"}) \vee vdo:S\_EH_1\_Value(?\text{z}, \text{"45diagonal"}) \wedge\\
&\\
&\quad vdo:hasCSD(?\text{x}, ?\text{a}) \wedge vdo:hasCSD\_Component(?\text{a}, ?\text{b}) \wedge\\
&\quad vdo:S\_DC_0\_Value(?\text{b}, \text{"Red-Orange"}) \wedge vdo:S\_DC_0\_Diff\_Value(?\text{b}, \text{"Medium"}) \wedge\\
&\quad vdo:S\_DC_1\_Value(?\text{b}, \text{"Yellow-Green"}) \wedge vdo:S\_DC_1\_Diff\_Value(?\text{b}, \text{"Low"}) \wedge\\
&\quad vdo:S\_DG_2\_Value(?\text{b}, \text{"Black"}) \wedge\\
&\quad \ldots.\\
&\textit{then}\\
&\quad ani:Cheetah(?\text{x});
\end{aligned}
$$

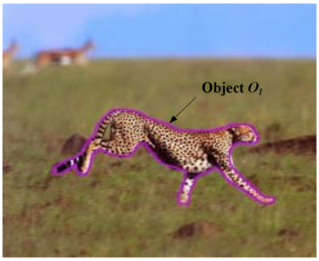


Fig. 5. The image including object O_1

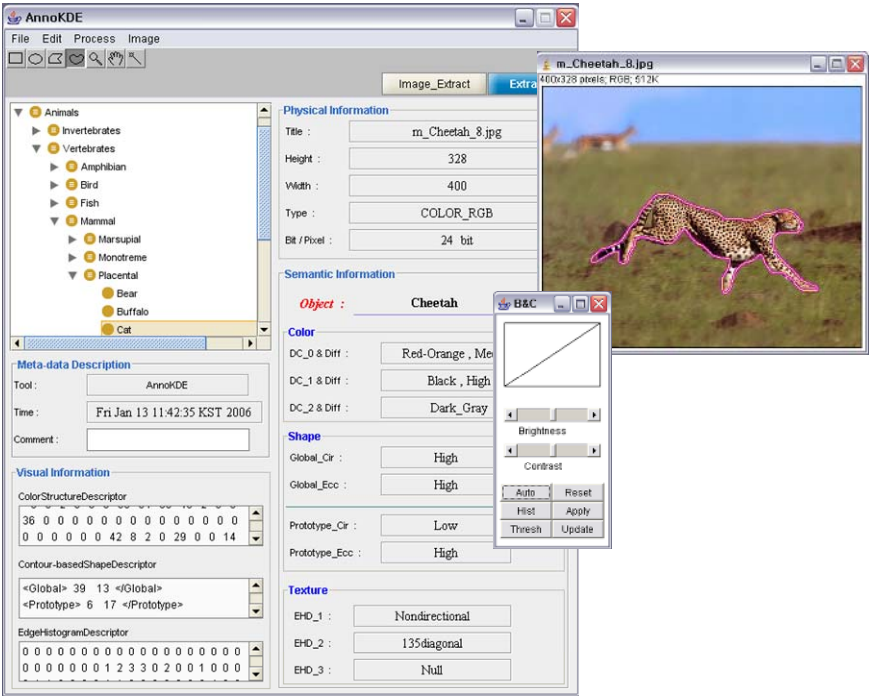
3. In order to infer object O_1, we apply *Rule_1*, which is written by ‘buchingae’ rule language, to the visual and animal ontology. In this case, object O_1 satisfies *Rule_1* because it has “*Nondirectional*”, “*135diagonal*” as the semi-concept values for the representative edge values and “*Red-Orange*”, “*Yellow-Green*” as the semi-concept values for the dominant color values, and so on. Therefore, object O_1 belongs to the *Cheetah* class in the animal ontology.
4. ‘Bossam’ rule engine supports various types of output format: RuleML, RDF, ATOML, etc. The following RDF statement shows the result of the semantic inference. In line 13~15, we can see that object *Object_1* (i.e., object O_1) has the ‘Cheetah’ class as its *rdf:type*.

```
1    <?xml version=″1.0″ encoding=″UTF-8″?>
2    <!DOCTYPE rdf:RDF[
3       <!ENTITY ns_1
        ‶http://kde.hanyang.ac.kr/ontology/Animals#″>
4       <!ENTITY ns_2
        ‶http://kde.hanyang.ac.kr/ontology/vdo#″>
5    ]>
6
7    <rdf:RDF
8       xmlns:rdf=″http://www.w3.org/1999/02/22-rdf
        -syntax-ns#″
9       xmlns:ns_1=
        ″http://kde.hanyang.ac.kr/ontology/Animals#″
10       xmlns:ns_0=
        ″http://kde.hanyang.ac.kr/ontology/vdo#″
11      xmlns:xsd=″http://www.w3.org/2001/XMLSchema#″>
12
13   <rdf:Description rdf:about=″&ns_0;Object_1″>
14      <rdf:type rdf:resource=″&ns_1;Cheetah″/>
15   </rdf:Description>
16   </rdf:RDF>
```

4 Design and Implementation

We have implemented the system, which links the high-level concept with the visual features by reasoning the object from the visual and animal ontology, using Java 1.4 and ImageJ 1.4 API which is the image processing tool. In particular, we have designed the user interface to enable users to see all of the image information more intuitively. The visual and animal ontology and the inference rules have been modeled by using Protégé 3.1 API and Bossam 8b30 API.

Figure 6 depicts the screenshot of the proposed system. The system can be functionally divided into 4 main parts: 1) Meta-data Description panel represents the metadata information, such as the *creation time* and *user comment*. 2) Visual Information panel describes the MPEG-7 visual descriptors of the corresponding image. 3) Technical Information panel represents the physical information of the corresponding image, such as title, height and width. 4) Semantic Information panel shows the semi-concept values and the high-level concept which is derived by applying the inference rules to the visual and animal ontology.



Meta-data Description

Tool : AnnoKDE

Time : Fri Jan 13 11:42:35 KST 2006

Comment :

Visual Information

ColorStructureDescriptor

```
36 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 42 8 2 0 29 0 0 14
```

Contour-basedShapeDescriptor

```
<Global> 39 13 </Global>
<Prototype> 6 17 </Prototype>
```

EdgeHistogramDescriptor

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 2 3 3 0 2 0 0 1 0 0 0
```

Physical Information

Title : m_Cheetah_8.jpg

Height : 328

Width : 400

Type : COLOR_RGB

Bit / Pixel : 24 bit

Semantic Information

Object : Cheetah

Color

DC_0 & Diff : Red-Orange , Me

DC_1 & Diff : Black , High

DC_2 & Diff : Dark_Gray

Shape

Global_Cir : High

Global_Ecc : High

Prototype_Cir : Low

Prototype_Ecc : High

Texture

EHD_1 : Nondirectional

EHD_2 : 135diagonal

EHD_3 : Null

Fig. 6. The screenshot of the system interface

First, users have to set the ROI from which they want to extract the semantic information and then press Extract button. Henceforth, their interaction is not necessary anymore. The image information, which consists of the high-level concept, physical information, visual descriptors, and metadata, is automatically extracted and described in the system interface. Finally, the image description which includes all the information of the corresponding image is generated as an OWL document importing the visual and animal ontology. The following image description is the OWL document representing the information for the corresponding image.

```
<Image rdf:ID="Image_1">
 <hasMeta>
  <MetaData rdf:ID="MetaData_1">
   <Tool rdf:datatype=http://www.w3.org/...>
    AnnoKDE</Tool>
   <Comment rdf:datatype=http://www.w3.org/...>
    This is a Cheetah image.</Comment>
                    ....
 </hasMeta>
 <hasAnimal>
  <ani:Cheetah rdf:ID="Object_1"/>
 </hasAnimal>
 <hasObject>
  <vdo:Object rdf:ID="Object_1">
```

```
14    <vdo:hasEHD>
15                       ....
16   </hasObject>
17   <hasVisualFeatures>
18    <ShapeDescriptor rdf:datatype=http://…>
19                       ....
20   </hasVisualFeatures>
21   </hasPhysicalInfo>
22    <PhysicalInfo rdf:ID=”PhysicalInfo_1”>
23     <Title rdf:datatype=http://www.w3.org/...>
24      m_Cheetah_8.jpg</Title>
25                       ....
26   </hasPhysicalInfo>
21  </Image>
```

Although our purpose is to automatically extract the high-level concept in an image, we have built the system generating the image description which includes the high-level concept as well as the visual features and physical information in order to satisfy various requirements of the users on the image retrieval. In line 17~20, MPEG-7 visual descriptors of the corresponding image is described. Also, since the users may only require the images which have the regular size or resolution in the image database, in line 21~26, the physical information of the corresponding image is represented.

5 Conclusions

Although CBIR is less time-consuming and provides more user-friendly access on images than the keyword-based approach, most users still use the keyword-based image retrieval to search the image that they want to find on the Web.

Ontology, which includes interrelationships among the concepts of a specific domain, helps to semantic annotations of images and facilitates image retrieval in a more convenient way. Also, ontology-based inference rules capture the expert knowledge and derive the new knowledge from knowledge existing in a specific domain.

In this paper, we proposed a novel method which automatically extracts the high-level concept from an image by building the visual and animal ontology, and then applying the inference rules to them in order to bridge the semantic gap. We first introduced the framework for mapping the MPEG-7 visual descriptors into the semi-concept values that human can understand easily, and also explained the visual and animal ontology that are built to bridge the semantic gap. The visual ontology facilitates the mapping between the visual features and the semi-concept values and allows the definition of relationships between the classes describing the visual features. Also, it is possible to apply the visual ontology for various domains, such as the cell images and other images domains. The animal ontology representing animal taxonomy is used to identify the high-level concept in the inference phase. We also described the method which infers the high-level concepts from images by defining the inference rules which map particular combinations of the visual features to high-level concepts defined in the animal ontology. In particular, such rules can be shared and collaboratively modified as the domain understanding shifts and changes.

The quality of MPEG-7 visual descriptors is very important in mapping of the visual features to qualified semi-concept value. Even though the image includes the homogeneous object, the MPEG-7 visual descriptors of the corresponding image may have different values. This leads to different semi-concept values for the homogeneous objects. Therefore, it is difficult to detect the inference rule from the training dataset which includes a specific object. In our experiment, we have detected six inference rules from the training dataset which consists of about 300 images including eight kinds of the animals. However, these inference rules are very useful for identifying the object in various images.

To extract rich semantic information from an image, we are also studying how to define the relationships among the dominant object and the background object of the image. For example, in the case of the semantic query such as “Lion at the zoo”, we have to define the relationship among the objects. Our future work will focus on addressing these problems.

References

[1] A. Pentland, R. Picard, and S. Sclaroff, “Photobook: Tools for Content-Based Manipulation of Image Databases”, SPIE Storage and Retrieval of Image & Video Databases II, 1994

[2] John R. Smith and Shih-Fu Chang, “VisualSEEk: a fully automated content-based image query system”, ACM Multimedia 96, 1996

[3] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, Jitendra Malik, “Blobworld: A System for Region-Based Image Indexing and Retrieval”, Third International Conference on Visual Information Systems, 1999

[4] Xingquan Zhu, Jianping Fan, Ahmed K. Elmagarmid, Xindong Wu, “Hierarchical video content description and summarization using unified semantic and visual similarity”, Multimedia Syst. 9(1), pp31-53, 2003

[5] A. T. Schreiber, B. Dubbeldam, J. Wielemaker, and B. J. Wielinga, “Ontology-based photo annotation”, IEEE Intelligent Systems, pp66-74, 2001.

[6] Shuqiang Jiang, Tiejun Huang, Wen Gao, "An Ontology-based Approach to Retrieve Digitized Art Images", Web Intelligence 2004, pp131-137

[7] Vasileios Mezaris, Ioannis Kompatsiaris, and Michael G. Strintz, "Region-based Image Retrieval using an Object Ontology and Relevance Feedback", EURASIP JASP, 2004

[8] Jianping Fan, Yuli Gao, Hangzai Luo, Guangyou Xu, "Statistical modeling and conceptualization of natural images", Pattern Recognition, 38(6), pp865-885, 2005

[9] ISO/IEC 15938-5 FDIS Information Technology. “MPEG-7 Multimedia Content Description Interface - Part 5: Multimedia Description Schemes”, 2001

[10] Spyrou E., Le Borgne H., Mailis T., Cooke E., Avrithis Y. and O'Connor N, "Fusing MPEG-7 visual Descriptors for image classification", International Conference on Artificial Neural Networks, (ICANN 2005), pp 11-15, 2005

[11] BS Manjunath, Philippe Salembier, Thomas Sikora, "Introduction to MPEG-7", wiley, 2002

[12] F. Mokhtarian and M. Bober, "The Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization", Kluwer Academic Publishers, 2002

[13] D. K. Park, Y.S. Jeon, C. S. Won and S. -J. Park, "Efficient use of local edge histogram descriptor", Proceedings of ACM International Workshop on Standards, Interoperability and Practices, Marina del Rey, California, USA, pp52-54, 2000

[14] Deborah L. McGuinness, Frank van Harmelen, "OWL Web Ontology Language Overview", W3C Recommendation, http://www.w3c.org/TR/owl-features/, 2004

[15] O. Lassila, R. Swick, "Resource Description Framework(RDF) Model and Syntax Specification", W3C Recommendation, World Wide Web Consortioum, 1999

[16] Hewlett-Packard, “Jena Semantic Web Framework”, http://jena.sourceforge.net/, 2003

[17] UMBC, “F-OWL: An OWL Inference Engine in Flora-2”, http://fowl.sourceforge.net/

[18] Gandon, F. L., Sadeh, N., “OWL inference engine using XSLT and JESS.”, http://mycampus.sadehlab.cs.cmu.edu/public_pages/OWLEngine.html

[19] Minsu Jang, Joo-Chan Sohn, "Bossam: An Extended Rule Engine for OWL Inferencing", RuleML 2004, pp128-138. 2003

Dental Decision Making on Missing Tooth Represented in an Ontology and Rules

Seon Gyu Park[1] and Hong-Gee Kim[2]

[1] College of Dentistry, Seoul National University
[2] DERI, Seoul National University
{ion2, hgkim}@snu.ac.kr

Abstract. The Web Ontology Language (OWL), which is a Description Logic based ontology language, is widely used to represent formal definitions of vocabularies for domain knowledge, especially in the medical domain. The Semantic Web Rule Language (SWRL), which is a Rule based ontology language, allows users to take advantage of inferencing of new knowledge from existing OWL knowledge base. In this paper, we describe a use case focused on building SWRL rule base on top of the tooth positional ontology represented in OWL so as to assist a dental decision-making on a missing tooth. Then, we discuss limitations of a current SWRL specification, through our experiences on converting dental knowledge into SWRL rules.

1 Introduction

Many dental procedures, such as mobile or hopeless teeth extraction, artificial teeth build-up, and root canal treatment, include the concept of 'missing'. The decision how a missing area is restored should be made on an everyday dental practice. However, this complicated judgment is the skill that can usually be gained by mentoring from other dentists, but not the knowledge that is educated by texts or an evidence-based learning. If cases are categorized in the reusable form of knowledge, dentists, especially the ones who lack experiences, will be able to get help for their decision making.

The Web Ontology Language (OWL) is a standard language for the purpose of declaring knowledge in a semantic web. OWL has three different flavors such as OWL-Lite, OWL-DL, and OWL-Full. Of these, OWL-DL is based on a Descriptive Language, powerful inter-connective semantic anchorage. Although OWL-DL is the best fit to the formalization of declarative knowledge in the medical domain, it is limited in expressing procedural knowledge such as time-oriented clinical decision. Thus, it is necessary to use a rule embedded OWL-DL in clinical decision-making. Besides, a rule is a very natural approach to formalizing medical experts' knowledge in that they usually solve a clinical problem on a procedural basis.

The Semantic Web Rule Language (SWRL) is proposed to enable users to formulate Horn-like rules in terms of OWL concepts. SWRL that extends the OWL abstract syntax can represent rules of the form of an implication between an antecedent (body) and a consequent (head). The intended meaning of an expression in SWRL is that whenever the conditions in the antecedent are satisfied, then the conditions in the

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 322 – 328, 2006.
© Springer-Verlag Berlin Heidelberg 2006

consequent must also be true. To improve common understandings, SWRL is suggested to be compatibility with a Semantic Web standard language, OWL. We illustrate a use case of dental decision making on missing teeth to combine OWL and SWRL expressions.

This paper presents a dental case study, calling for reasoning with an OWL-DL ontology and SWRL-based rules. This works toward helping decision of restoring a missing tooth. On an ontology part, we describe a positional relation with a specific tooth. On a rule part, we aim at looking for an abutment of the restoration that will be made in the process of relating "neighbor" to "missing tooth".

2 Ontology

Adult humans normally have 32 teeth spreading over each quadrant evenly. Each quadrant of 8 teeth consists of these types of teeth: central incisor, lateral incisor, canine, first premolar, second premolar, first molar, second molar, third molar. The third molar commonly refereed to as wisdom tooth may or may not erupt. Incisors and canine are front teeth that usually capture one's appearance, so both patients and dentists should pay attention to the cosmetics for the treatment. The premolar and the molar that are located in the back of the mouth play important roles in grinding up foods. Since chewing may invoke a lot of stress to these teeth, the strength is a very important concern in the replacement of these teeth.

In our ontology we modularize these concepts to keep these simple in terms of maintainability, reusability, and evolution ability [1]. We set a few different classes according to the usability but these concepts may or may not be equal. We distinguish the tooth names into general names and specific names in that the former contains only the positional information of the tooth for general purposes whereas the latter is directly used for dental decision making for treatment. The "SpecificName" classes such as "CentralIncisor" and "FirstPremolar" specify the functional roles of the tooth. A need for classifying a specific tooth forces us to define another class, "ToothNumber", which may be considered as redundant at a concept level. For an example, "w11" in "ToothNumber" is identical to "CentralIncisor" in "SpecificName". Inversely, "CentralIncisor" is the same as "w11", "w21", "w31", and "w41" because "CentralIncisor" doesn't represent if it is located in a right or left, as well as a upper or lower.

In order to identifying a specific tooth whether it is positioned left or right, a standard way of naming the specific tooth is necessary. There are various different dental notation systems for identifying a specific tooth. There are three international standard systems for naming teeth: the universal numbering system, the palmer notation method, and the two-digit FDI world dental federation notation. In this paper, we follow two-digit FDI world dental federation notation that is also known as ISO-3950 notation. It provides a system for designating teeth or areas of the oral cavity using two digits. We assert these notations into "ToothNumber" class.

Many dentists, sometimes, hesitate to choose which abutments are useful because a lot of alternative treatments exist according to the missing tooth numbers or positions. In the case of missing teeth in the anterior region, a dentist should keep in mind of cosmetics. On the other hand, missing teeth are located in the posterior region the dentist considers biomechanics so as to satisfy a patient's chewing efficiency. Thus, a

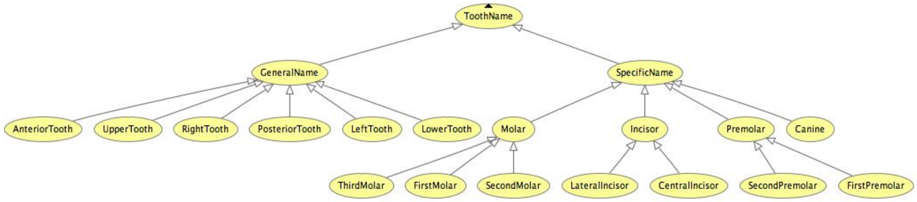


Fig. 1. The class hierarchy

3-dimensional tooth position should be declared so that the reasoning over knowledge base of tooth position can be processed. The declaration of the tooth position assists a proper decision of missing teeth case.

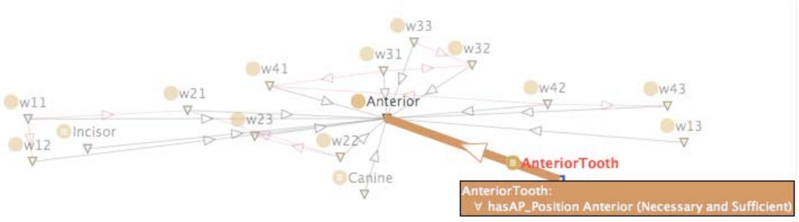


Fig. 2. ‘hasAP_Position’ property

On our ontology part, we set up the following three kinds of relations: “hasAP_Position”, “hasRL_Position”, and “hasUL_Position”. These relations or properties take their ranges with their positional values from the “Positional_Value_Partition” class which has 6 subclasses such as “Anterior”, “Posterior”, ”Right”, “Left”, “Upper”, and “Lower” classes. The property of “hasAP_Position” has its range value from either “Anterior” or “Posterior”. The property of “hasRL_Position” property has its range value from either “Right” or “Left”. The property of “hasUL_Position” has its range value from either “Upper” or “Lower”.

For an instance, “AnteriorTooth” class has a “hasAP_Position” property which has its property value as “Anterior” class. It is very advantageous to put a class such as “Anterior” as a property value because the “Anterior” class can consistently tell us the position located at the front, and it get used again and again easily. This complicated work is done by using protégé wizards plug-in. [2] Another approach, which is more intuitive to an ontology engineer, considers values as sets of individuals. However, we cannot have further branching of values because an individual cannot have a sub-individual. Moreover, no reliable inference appears to popular reasoners such as FaCT and Racer because those reasoners cannot completely handle with all required reasoning over individual. For those reasons, we make a positional value as a class that must have a single individual.

The relationship of neighborhood is defined by a symmetric property, “hasNeighbor”. This property is just described in “ToothNumber” classes, since “ToothNumber” classes indicate specific teeth of an adult. In this paper, we are not concerned with children’s teeth not only because a decision making of rehabilitation of missing tooth is not usually applied to children's teeth, but also because the Universal Numbering system is equivalent to FDI World Dental Federation notation.

In the class hierarchy, we put all useful general concepts into ontology. For example, the concept "UpperTooth" helps a user to look for a proper abutment in the case of missing two neighbored upper central incisors. The missing case of two neighbored lower central incisors is different form that of two neighbored upper central ones in that the former case requires a single abutment whereas the latter case requires two abutments in order to build biomechanically a more reliable fixed partial denture.

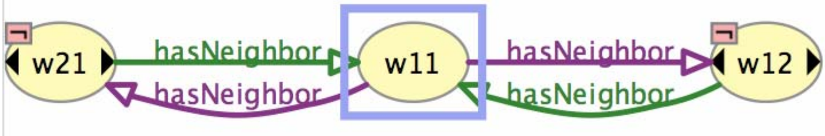


Fig. 3. 'hasNeighbor' property

3 Rule

An approach to using a rule on an OWL knowledge base is to represent OWL's properties and classes in SWRL within the DL's boundary. This approach has a few advantages for maintaining and utilizing OWL's semantics and syntax, while the expressivity is limited [3].

An alternative approach is the one that extends propositional logic or Description Logic like OWL-DL into First order logic or Logic programming when using a rule. Recent proposal of the semantic web rule language, WRL (Web Rule Language) is an example [4]. In the WRL proposal, there is an intersection such as DLP fragment between First order logic and Logic Programming so that WRL becomes more expressive but a complicated reasoning methodology may be needed in the case of using a descriptive logic reasoner.

In this paper, we use SWRL for reasoning over OWL knowledge base since SWRL has a fully compatibility with OWL. In addition to the undecidable feature of SWRL, using OWL and SWRL in a single context is very comfortable to an ontology engineer due to the syntactic and semantic homogeneity of the two languages.

On a SWRL specification, it can reason with an individual. The first thing we need is to assert an individual to an OWL class. An individual "w13_1", for instance, can be a member of the class, "w13", which indicates an upper right canine. Canine is located at a very strategic position where a dentist constructs a fixed partial denture. It has a very strong root and is positioned at a turning point of a dental arcade, so the replacement of "w13" is a very difficult decision to a dentist. The problem arises in the situation that a dentist should make a decision of whether an abutment is used, or how many abutments can be utilized. Therefore, a rule should be dependent upon a concept where a domain expert, a textbook, or a journal article describes what property a missing tooth has. It is safe to say that a well-made rule is very comprehensive to a dentist when the modification is needed.

The following is a general rule for determining which tooth is a candidate for an abutment of a fixed partial denture. An available tooth adjacent to a missing tooth is usually the most suitable candidate for an abutment. It should not be a missing tooth owing to the fact that the abutment can be utilizable to a dentist when creating a fixed partial denture. Thus, we need to inspect whether the neighborhood of a missing tooth is missing or not. Before building the following rule, we should define "Remaining-Tooth" as a subclass of "ToothNumber" in an OWL knowledge base, excluding "MissingTooth" since there is no way to represent *negation-as-failure* if the restricted form of the SWRL rule is not extended. [1]

$$\text{MissingTooth}(?x) \wedge \text{hasNeighbor}(?x, ?y) \wedge \text{RemainingTooth}(?y) \rightarrow \text{Abutment}(?y) \wedge \text{Pontic}(?x)$$

In fact, SWRL is introduced in order to overcome the limitation of OWL which is impossible to explain the relationship between properties. The basic example is the "uncle" property comprising of the "parent" and "brother" properties [5]. Beside of telling the connection between properties, SWRL is useful to assert an individual to the specific class that did not include the individual when the condition was not full-filled. The above example shows that an individual of "MissingTooth" class goes to "Pontic" class when the body part of the rule is true. With this 'syntactic sugar', SWRL provides us with ways of efficient ontology making, a straightforward understanding, and easy manipulation when a change is needed.

Then, we have made more specific rules that concern a single tooth missing, the missing of two neighbor teeth, an anterior tooth missing, and a posterior tooth missing, and so on. The recommendation for the decision of replacing missing teeth can be represented in rules in accordance with the textbook of prosthodontics [6]. For instance, if two continuous teeth are missing, abutments as their very next neighborhood is not sufficient because of biomechanics. Periocemental spaces of abutments exceed their pericemental space of missing area. There are more complicated cases other than just two neighboring teeth missing case.

$$\text{w11}(?x) \wedge \text{w21}(?y) \wedge \text{MissingTooth}(?x) \wedge \text{MissingTooth}(?y) \wedge \text{w13}(?ab1) \wedge \text{w23}(?ab2) \rightarrow \text{Abutment}(?ab1) \wedge \text{Abutment}(?ab2)$$

The above example shows that missing two anterior teeth allows a dentist to use double abutments meaning that we have to build 6-unit fixed partial denture undoubtedly. The main reason for it is to give a more reliable artificial denture to patients, and a further explanation is well described in [6].

4 Discussion

In addition to the discussed approach, there may be other approaches to representing rules in the semantic web. RuleML [2] is an XML basis and extensive coverage. RuleML provides an extensive lattice of sublanguages ranging from production rules to regular Logic Programming to First-order logic, as well as sublanguages with such features as slotted syntax, meta-programming, and courteous logic programming. Datalog, naf (Negation-as-failure) or First order logic are examples. A major advantage of RuleML is flexibility owing to the fact that it has possibility to extend.

[1] http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/#7.1

[2] http://www.ruleml.org

SWRL is based on top of the DL species of OWL, so that backward compatibility is more efficient than any other proposals. However, Non-monotonic reasoning is not possible because SWRL per se is FOL basis. Moreover, at the point of ontological engineering view, SWRL restricts several syntactic expressions, for an instance, negation-as-failure. In our use case, we need a negation in rule syntax since the opposite concept of “MissingTooth” is able to verify the abutment candidate. Instead, we should have made the “RemaingTooth” class which means a negation of “MissingTooth” just in “ToothNumber” class because of a syntactic restriction of SWRL.

Adding a temporal concept to the concept of “MissingTooth” either in a rule base or an ontology base conveys impossibility in terms of syntactic limitations since predicates with an n-arity is not enabled in SWRL or OWL. Therefore, we would nee to assert a temporal concept to “MissingTooth” class in a snapshot.

In [7] Horrocks *et.al.* points out that a to-be-standard rule should have a maximum backward compatibility with OWL which is a standard ontology language. They also argue that a general translation from ontology base to rule base is very expensive if ontology base already made by OWL or RDF extends into a lot different kinds of logics, ‘two tower’ approach in this literature. Even in consistency, we need to see if the RDFS/OWL and the Rules are computing the same thing. Some way to check consistency between some or all of the rules and ontologies is needed. Despite of being a ‘two tower’, the restriction of syntax and logic in the purpose of backward compatibility carries some troubles in terms of building ontology. In resolving the reasoning problem, hybrid approach through scoped inference is suggested. [8]

Throughout our experience of building rules and an ontology, the expression of negation-as-failure as well as predicate with more than 2 arity in a rule is needed. In addition, there is a use case of identifying the part-whole relation in brain cortex structure using a rule [9]. This use case represents the boundary of brain cortex with OWL and SWRL for helping label the brain cortex structure. It shows a rule is very efficient and effective way to engineering ontology. We insist that a rule is more effective means to transport medical knowledge from a domain expert than a description because a rule is in general intuitive to read and understand. Accordingly, the representation of some kinds of medical knowledge in a rule is interpretable, manageable, and feasible.

5 Conclusion

Identifying Tooth position in 3 dimensions is resolved by OWL property having positional value as class. A rule is a very natural way to convey domain expert’s knowledge into knowledge base.

Through our experience, SWRL specification has some limitation when building a rule such as negation-as-failure and predicate with n-arity. Further studies are needed to make a new standard rule language. Rule Interchange Format Working Group was founded recently. This working group is chartered to yield an interoperable rule language.

Acknowledgement

This study was supported by a grant of the Interoperable EHR Research and Development Center (A050909), Ministry of Health & Welfare, Republic of Korea.

References

1. Rector, A. : Modularisation of Domain Ontologies Implemented in Description logics and related formalisms including OWL. In Knowledge Capture 2003, (Sanibel Island, FL, 2003), ACM, 121-128
2. Rector, A. Drummond N. Horridge M. Rogers J. Knublauch H. Stevens R. Wang H. Wroe C. : OWL Pizzas: Common errors & common patterns from practical experience of teaching OWL-DL. In European Knowledge Acquisition Workshop (EKAW-2004), 2004.
3. Horrocks, I. Patel-Schneider, P.F. Boley, H. Tabet, S. Grosof B. and Dean, M. : SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21 May 2004. Available from http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/
4. Angele, J. Boley, H. Jos de Bruijin Fensel, D. Hitzler, P. Kifer, M. Krummenacher, R. Lausenn H. Polleres, A. Studer, R. : Web Rule Language(WRL). W3C Member Submission, 09 September 2005. Available from http://www.w3.org/Submission/WRL/
5. Horrocks, I. Patel-Schneider, P.F. : A Proposal for an OWL Rules Language. Proc. of WWW 2004, ACM, 723-731
6. Shillingburg, H.T. Hobo, S. Whitsett, L.D. Jacobi, R. Brackett, S. : Fundamentals of fixed prosthodontics 1997, Quintessence publishing
7. Horrocks, I. Parsia, B. Patel-Schneider, P.F. Hendler, J. : Semantic Web Architecture: Stack or Two Towers? In Third Workshop on Principles and Practice of Semantic Web Reasoning, Dagstuhl, Germany, September 2005
8. Kifer, M. Jos de Bruijin Boley, H. Fensel, D. : A Realistic Architecture for the Semantic Web. In RuleML Conference proceedings, Galway, November 2005
9. Golbreich, C. Bierlaire, O. Dameron, O. and B. Gibaud. Use case: Ontology with rules for identifying brain anatomical structures. W3C Workshop on Rule Languages for Interoperability, 2005

Ontology Driven Visualisation of Maps with SVG – Technical Aspects

Frank Ipfelkofer, Bernhard Lorenz, and Hans Jürgen Ohlbach

Institute for Informatics, Ludwig-Maximilians University, Munich
fipfelkofer@kastner.de, {lorenz, ohlbach}@pms.ifi.lmu.de

Abstract. This article describes the technical aspects of a particular use of ontologies for visualising maps in a browser window. Geographic data are represented as instances of concepts in an ontology of transportation networks which was designed in close relation to GDF. These data are transformed into SVG, whereas the transformation is specified symbolically as instances of a *transformation ontology*. This approach is extremely flexible and easily extendible to include all kinds of information in the generated maps. This article is an abbreviated version of the research report [5] and focuses only on the technical aspects of the approach.

Keywords: semantic web, ontologies, visualisation, svg, semantic techniques.

1 Introduction

There are may different ways for generating maps as images on a computer. The most straightforward way is to read the geographic data from a data source and to use special purpose algorithms that transform the data into some bitmap graphics format. These algorithms – as well as the whole process – are rather complex and not easy to change or extend. Only experts who are familiar with the details of the process can do this. The algorithms depend very much on the particular data format and they usually yield only static pictures.

An alternative method is to generate output by means of ontologies and ontology instances instead of using specialised algorithms processing data transformed into generalised formats. Furthermore, instead of creating bitmap graphics, the results are encoded in a graphics description language, such as Scalable Vector Graphics (SVG), which is used as an example throughout this article. Several advantages result from this approach which are briefly laid out in this introduction and discussed in more detail in section 2.

Scalable Vector Graphics (SVG) [7] is an XML-based language for describing geometric objects. Compared to bitmap graphics, SVG has a number of advantages, which are discussed in detail in [5]. Within the scope of this work, the most important property of SVG is the possibility to manipulate an SVG document via the underlying DOM representation, thus enabling interaction through the browser's SVG plugin. Furthermore, SVG is device- and renderer

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 329–335, 2006.
© Springer-Verlag Berlin Heidelberg 2006

independent and SVG data can be manipulated easily using generic algorithms available for common vector data.

The primary data sources for the visualisation are usually Geographical Information System (GIS) databases which (in-)directly provide data in standard formats, for example the Geographic Data Format (GDF) [2] or the Geography Markup Language (GML) [3]. This is not the only choice. In this article we propose an alternative. We still use GIS data in some of the standard formats as primary source, but only because these are the only available data. The idea is to take an OWL ontology of transportation networks to represent the data as instances of the concepts of this ontology. OWL provides a data format for instances of the concepts of the ontology (basically the Resource Description Framework (RDF)), and we use RDF for the GIS data. This is not just a syntactic reformulation. It offers completely new possibilities because the data format is only loosely coupled with the ontology. For example, consider an ontology containing the concept of a *road*. A road may have directions, at most two. If there is a particular road R with directions = 1 then OWL would classify R as a *road*. If, in a later step, the ontology is extended with the concept *one_way_road* as a *road* with directions = 1 then OWL would automatically reclassify R as a *one_way_road*. Thus, there is a certain degree of independence between the OWL data format and the ontology. The same data can be used for different ontologies.

We developed the **Ontology of Transportation Networks** (OTN) [1]. OTN was generated by making the concepts and structures which are implicitly contained in GDF explicit as an OWL ontology. OTN contains all kinds of notions for transportation networks, roads, trains, ferries and much more.

The method for visualising maps proposed in this article is as follows[1]:
There are three classes of input data

1. Concrete GIS data which has been transformed into the OWL data format (we used the map of the city of Munich).
2. An ontology of transportation networks, in our case OTN.
3. A set of transformation rules which determine how the instances of the concepts in the ontology are to be transformed into SVG code.

The transformation algorithm now applies the transformation rules to all relevant instances of the concepts in the ontology and produces SVG documents as output. With this architecture it is extremely easy to change the visualisation. For example, if we want to distinguish one-way roads from ordinary roads, it is only necessary to introduce the concept of a one-way road in the ontology and to add a corresponding transformation rule.

In this document we focus on the technical aspects of our approach, a more detailed description can be found in [5,4]. Section 2 shortly lays out the limits of SVG and the functionality of our extensions. The transformation step is dealt with in section 3, before we conclude this paper with a short summary in section 4.

[1] This method was also implemented, see [4].

2 SVG Visualisation

The following description of the SVG visualisation is but a brief overview to motivate some of the design decisions for the transformation method. A detailed description is not within the scope of this article and can be found in [5].

The final result of the visualisation is illustrated in fig. 1. The browser window consists of a frame containing the SVG map and a HTML menu on the right hand side. The SVG map is zoomable in a wider range than the built-in SVG zooming facility would allow. The map may also contain dynamic elements (e.g. buses or trains moving along the rails, clouds moving over the scene, etc.).

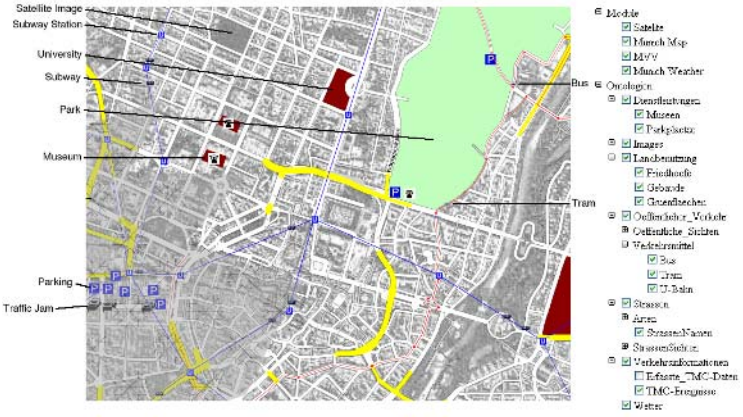


Fig. 1. Visualisation in SVG

The menu allows the user to choose what he wants to see. It is divided into two main sections, *Modules* and *Ontology*, whereas the former represents modules in the sense of different input media, such as rasterised data (satellite images), vector data (street network), and specialised data sources (e.g. dynamically generated weather data or data about the public transport system) which require a taylor made plugin. The latter consists of the instances of OTN.

Three features of the system extend the SVG capabilities: *dynamic loading*, *rasterisation of data*, and *zoom mechanism*. The monolithic nature of an SVG document entails that a document can only be loaded as a whole. This means either short loading times and limited data to view, or respectively longer loading times as the amount of data increases. This is in contrast to typical user behaviour on the web, as the user expects (first) results to be displayed almost immediately, while more data can be loaded on the go while browsing the site. The **dynamic loading** mechanism of our approach facilitates the quick loading of a map and dynamically loads subsequent parts of the map as the user pans or zooms into the map.

As the dynamic loading mechanism relys on the map being split into an number of smaller data chunks, the question of **rasterisation** arises. In order to produce files of roughly equal size, a special rasterisation of the vector data is done on the server side. Because a naive algorithm would produce larger files in densly populated areas and smaller files in other areas, a simple chessboard approach did not suffice. Instead, we make use of an R-Tree [6] structure to produce tiles of equal content, covering areas of different size.

Since SVG does not provide **zooming capabilities** powerful enough for a map application, which means wide range zooming from large scale (countries) to small scale (side streets and pedestrian zones), another extension was needed. On one hand, very small structures (e.g. names of small streets) shall not be displayed at a large scale (an therefore not even be loaded at a certain stage). On the other hand, zooming should be possible over a much wider range than the built in 16-fold zoom of SVG documents. Our approach facilitates these two techniques.

3 From OTN to SVG

In a preparatory step the GIS data has been transformed into the OWL data format of the OTN ontology. All information about roads, bus lines, underground lines, parks, etc. are therefore stored as instances of the OTN ontology. In order to generate the many little SVG files which contain the tiles of the map at the various zoom levels, one could now write a bulky program that reads the map and somehow generates the SVG files. This would be extremely complicated and inflexible. Therefore we took another route.

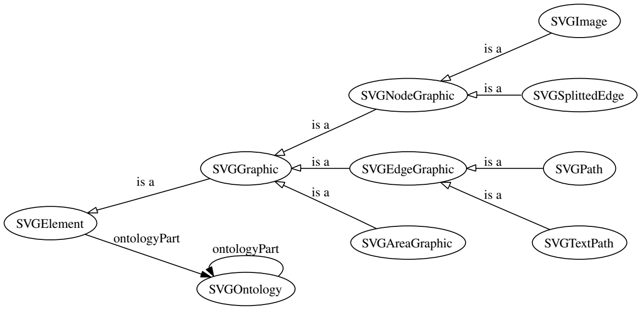


Fig. 2. Transformation ontology for transformations from OTN to SVG

SVG has a relatively small fixed number of constructs for displaying graphical structures. These few constructs are also represented as an OWL ontology, the *transformation ontology*. The main parts are depicted in fig. 2. The *SVGOntology* class does not correspond to an SVG construct. It defines the structure of elements in the SVG document which are to be displayed under "ontology" in the menu on the right hand side of the browser window (see fig. 1). Each component of the map is to be transformed into one of these SVG elements. For example, a road may be transformed into an SVG path element. A railway or a bus line may also be transformed into an SVG path element. The idea is now to generate an instance of the corresponding class of the transformation ontology for each element of a map that is to be transformed into a particular SVG element. This instance must contain the information *how to* transform the map element into SVG. To illustrate this, consider the following instance of SVGPath:

```
<SVGPath rdf:ID="BusLine">
  <useOnClass>Route_Link</useOnClass>
  <condition>=[public_Transport_Mode]=Bus</condition>
  <minDetail>0</minDetail><maxDetail>40000</maxDetail>
  <paintingOrder>300</paintingOrder>
  <width>3</width>
  <groupAttributes>class="Bus"</groupAttributes>
  <elementType>path</elementType>
  <addId>false</addId>
  <ontologyPart rdf:resource="#oeffentl_Verkehrsnetz_ontologyPart"/>
  <ontologyPart rdf:resource="#Bus"/>
</SVGPath>
```

It specifies how the OTN data `Route_Link` with `public_Transport_Mode` = `Bus`, which represents a segment of a bus line, is to be transformed into an SVG path element. The important parts are `<useOnClass>Route_Link</useOnClass>` and `<condition>=[public_Transport_Mode]=Bus</condition>`. It means that the transformation is to be applied to all instance of the class `Route_Link` which satisfy the condition `public_Transport_Mode=Bus`. The elements `minDetail` and `maxDetail` specify the zoom level for which this transformation is to be applied. The remaining elements of `SVGPath` specify geometric and other details to be inserted into the SVG path element. The actual coordinates for the path element are directly taken from the OTN data.

In the next example we want to put a small moving image of a bus onto the SVG path element of a bus line. SVG has features for generating dynamic graphics. Unfortunately it turned out that in the currently available browsers they slow down the rendering so extremely that they are just not usable. Therefore the system generates moving images on a map by periodically downloading a new version from the server. This is specified in the next example.

```
<SVGImage rdf:ID="Bus">
  <useOnClass>Line</useOnClass>
  <condition>=[public_Transport_Mode]=Bus</condition>
  <minDetail>0</minDetail><maxDetail>40000</maxDetail>
  <url>images/bus.gif</url>
  <updatePeriod>5</updatePeriod>
  <xCoord>=[x]-15</xCoord><yCoord>=[y]-25</yCoord>
  <height>50</height><width>30</width>
  <onClick>=IF [external_Link] THEN
    window.top.open ("[external_Link]")</onClick>
  <tooltip>=Bus|Linie [alternate_Name] |
    Departure Time: {TIME(3)@[startTime]} \- [starts_at].[ID] |
    Arrival Time: {TIME(3)@[endTime]} \- [ends_at].[ID] |
    Waiting Time: {TIME(2)@[waitingTime]} |
    Travel Time:  {TIME(2)@[drivingTime]}</tooltip>
  <ontologyPart rdf:resource="#Bus"/>
  <ontologyPart rdf:resource="#aktueller_Betrieb_ontologyPart"/>
  <paintingOrder>10000</paintingOrder>
  <addId>false</addId>
  <viewbox>-30 -30 25878 23419</viewbox>
</SVGImage>
```

This time we use an SVGImage element to insert the image images/bus.gif into the map. The transformation is to be applied to OTN instances of Line with attribute public_Transport_Mode=Bus. In order to update the SVG file every 5 seconds, the update period is set as <updatePeriod>5</updatePeriod>. If the image is to be moved then this file must be updated at server side in the same regular intervals. The generated SVG code would look like this:

```
<g  ontology="aktueller_Betrieb Bus">
<!-- Start of LOADNODE -->
<image onclick='if (window.top.ALLOW_ONCLICK){window.top.open
   ("http://efa.mvv-muenchen.de/mvv/XSLT_TTB_REQUEST?lineName=54")}'
   x='17763.23'  y='10898.37'  width='30'  height='50'
   xlink:href='images/bus.gif'  onmouseover='TOOLBAR.Show(evt)'  >
   <title>Bus<BR/>Linie 54
             <BR/>Abfahrt: 20:38 - Mauerkircherstrasse
             <BR/>Ankunft: 20:40 - Herkomerplatz
             <BR/>Haltedauer 00:00:15
             <BR/>Fahrtzeit 00:01:45</title>
</image>
<!-- End of LOADNODE -->
</g>
```

Putting it all together. Now we have the data source, i.e. the GIS data as OTN instances in the OWL format. We have the SVG graphics elements as the transformation ontology in OWL, and we have transformation rules as instances of the transformation ontology. This is the declarative part. The actual transformation is now done by a particular Java program. For each element of the transformation ontology (see fig. 2) there is a corresponding Java class. They have methods which know how to match the OTN data with instances of the transformation ontology and how to generate SVG code from this.

For example, there is a Java class SVGImage. This class can be instantiated with the parameters of the SVGImage instances of the transportation ontology, the Bus instance from above, for example. Now we have a Java object whose methods are able to search through the OTN data and to identify the items for which SVG code is to be generated that inserts the symbol for the bus. This information is inserted into an R-Tree, and from the R-Tree the system generates the SVG files for the tiles of the map. The fact that the transformed data need to be grouped with an R-Tree makes simpler approaches, for example via XSLT, much more difficult.

4 Summary and Outlook

In this work we illustrate a particular use of ontologies for dealing with geographic data which makes it possible to adapt the ontology to the needs of the application and still work with the same data. The transformation of the geographic data into SVG is also controlled by an ontology as the SVG elements are represented as concepts of a *transformation ontology* and the particular rules for transforming the data in a particular way are specified as instances of the concepts of the transformation ontology. By changing these instances or creating new instances one can change or extend the displayed maps very easily. The integration of dynamic or temporal information, e.g. to display real time data and/or changes over time is one of several possible extensions of this work.

Acknowledgement

This research has been co-funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779 (cf. `http://rewerse.net`).

References

1. Bernhard Lorenz and Hans Jürgen Ohlbach and Laibing Yang. Ontology of Transportation Networks. REWERSE Deliverable A1-D4, University of Munich, Institute for Informatics, 2005.
2. International Organisation for Standardisation (ISO). Intelligent transport systems - Geographic Data Files 4.0 (GDF) - Overall data specification, ISO/DIS 14825/2004, February 2004.
3. Geography Markup Language GML, Version 3. `http://www.opengis.org/docs/02-023r4.pdf`, (accessed 11/2005).
4. Frank Ipfelkofer. Basisontologie und Anwendungs-Framework für Visualisierung und Geospatial Reasoning. Diploma thesis, University of Munich, Institute for Informatics, 2004.
5. Frank Ipfelkofer, Bernhard Lorenz, and Hans Jürgen Ohlbach. Ontology Driven Visualisation of Maps with SVG – An Example for Semantic Programming. Forschungsbericht/research report PMS-FB-2006-5, Institute for Informatics, University of Munich, 2006.
6. R-Tree Portal. `http://www.rtreeportal.org`, Juni 2003.
7. Scalable Vector Graphics (SVG) 1.1 Specification, W3C Recommendation. `http://www.w3.org/TR/SVG11`, January 2003.

Applying CommonKADS and Semantic Web Technologies to Ontology-Based E-Government Knowledge Systems

Dong Yang, Lixin Tong, Yan Ye, and Hongwei Wu

Department of Industrial Engineering, Shanghai Jiao Tong University,
200030 Shanghai, China
dongyangcn@hotmail.com, culizn@163.com, yeyan_yan@yahoo.com.cn,
hom@sjtu.edu.cn

Abstract. Government agencies are the largest owners of knowledge assets such as regulations, documents, forms. To build a knowledge-based system (KBS) for e-government has proved to be an effective way to enhance the efficiency of handling governmental services. However, few efforts are made to address automatic reasoning of knowledge-intensive tasks within e-government processes. For this purpose, we present an approach to building an e-government KBS by using the CommonKADS, a knowledge-engineering methodology, and semantic web technologies (OWL, SWRL, OWL-S), with the aiming of automatically solving knowledge-intensive tasks within e-governmental services. Our experiences show that the CommonKADS is crucial to the analysis and identification of knowledge-intensive tasks within government processes, whereas the semantic web technologies enable the refinement of domain ontologies, domain rules and task methods.

1 Introduction

Electronic government aims to enhance efficiencies of government services and reduce operational costs by means of ICT (information and communication technologies) [1]. Currently, more and more countries have undertaken to implement e-government strategies ranging from government websites to e-democracy [2]. Special emphases are placed on the application of knowledge management to electronic government as government agencies are the largest owners of knowledge assets such as regulation, documents, legislation, forms, etc. Governmental processes for offered services are characterized by their heavy dependencies on various knowledge such as regulations and rules to solve problems [3]. From the viewpoint of knowledge engineering [4], they belong to knowledge intensive processes. Automation of the knowledge intensive tasks within these governmental processes can significantly reduce the efforts in handling cases, thereby improving working efficiencies of government agencies. To enable automatic reasoning of knowledge-intensive tasks, building a knowledge-based system for e-government is crucial. The knowledge engineering methodologies offer such a way to construct a knowledge-based system by

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 336 – 342, 2006.
© Springer-Verlag Berlin Heidelberg 2006

modeling domain knowledge, rules knowledge and inference mechanisms. CommonKADS, one of the commonly used knowledge engineering methodologies, provides a complete framework for building a KBS system, ranging from process analysis and knowledge acquisition to knowledge modeling [4]. Nevertheless, many new technologies such as the semantic web technologies have not been covered in these classic methodologies. We argue in this paper that these classic knowledge engineering methodologies are still effective in the earlier phases of constructing an e-government KBS, such as process analysis and knowledge acquisition, whereas semantic web technologies have advantages in ontology modeling and refinement. As a result, we can take advantages of both CommonKADS and semantic web technologies to build a knowledge-based system to support the automatic reasoning of knowledge-intensive tasks within e-government processes.

This paper is organized as follows. A framework for constructing an e-government KBS that supports automatic reasoning of knowledge intensive tasks is given in section 2. Section 3 discusses the identification of knowledge through the commonKADS models. Section 4 describes knowledge acquisition from e-government regulations with the help of the PC-PACK tool. In section 5, OWL is employed to model e-government domain ontology with the support of Protégé-OWL tool. In section 6 and 7, the rule and task knowledge within e-government are formally captured in SWRL and OWL-S, respectively. Section 8 concludes this paper.

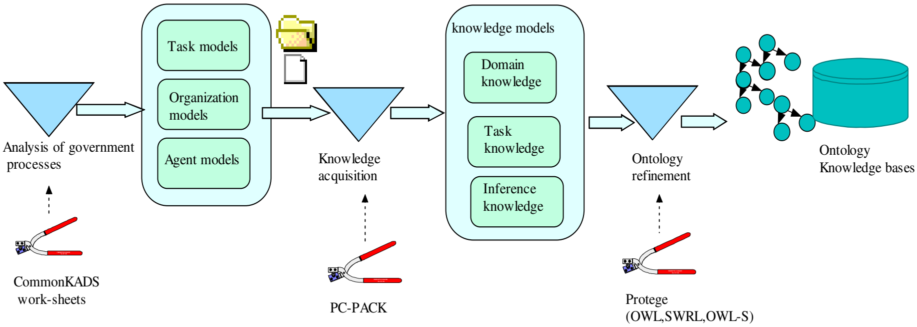


Fig. 1. A framework for developing an e-government KBS

2 A Framework for Developing an E-Government KBS

To start with analysis of business processes, the CommonKADS knowledge engineering methodology uses a suit of models to identify knowledge-intensive tasks and the knowledge items on which these tasks depend [4]. The models that are in the form of a set of worksheets consist of organization models (OM1~4), task models (TM1~TM2) and agent models (AM-1). Then, the concepts and rule knowledge that constitute knowledge models are extracted from these knowledge items through the use of knowledge acquisition techniques. In the CommonKADS, however, the concepts are defined with CML (Conceptual Modelling Language) [5], a semi-formal language. Instead of using CML, we choose OWL (Ontology Web Language)[6] to formally define the domain ontology. The advantage for using OWL is that the concept consistencies can be checked due to its description logic based semantics. Further, compared to CML, a wide range of existing tools have been available for manipulation of OWL ontologies without the burden of developing specific tools. Combining the CommonKADS and semantic web technologies, we present a framework for developing an e-government KBS, as shown in figure 1. The procedures are classified into four phases:

1) Identifying knowledge-intensive tasks and knowledge items

According to the CommonKADS analytical approach, both the knowledge-intensive tasks within e-government processes and the knowledge items that these tasks make heavy use of can be identified through the use of task models (TM1~TM2) and organization models (OM1~4).

2) Knowledge acquisition

For identified knowledge items (such as regulation and rules) that are in the form of electronic documents, database, etc., knowledge acquisition techniques such as interview, protocol analysis, concept sorting are utilized to extract concepts and knowledge from these items. PC-PACK tool [7] can be used to facilitate the process of knowledge elicitation.

3) Ontology refinement and modeling

Refinement on domain concepts identified in the above phase needs to be carried out to organize these domain concepts in a semantically structural way. Then OWL is employed to formally represent e-government domain ontology. Additionally, rule and task knowledge e-government are expressed in SWRL [9] and OWL-S [8], respectively.

4) Implementation for a KBS

In this phase, a rule engine is chosen to make inferences based on the matching of facts with rules. A knowledge engine is developed for the coordination of the steps (inferences) used to solve a knowledge-intensive task.

3 Knowledge-Intensive Tasks and Knowledge Items

According to the CommonKADS analytical approach, we adopt organization models (worksheets OM1~OM5) and task models (worksheets TM1~TM2) to analyze this process aiming to identify knowledge-intensive tasks within it. A set of worksheets are completed by interviewing with officers from these agencies. Table 1 shows a task model TM1 for describing the task "assess social security card (SSC) application" within e-government domain. In accordance with this task model, the knowledge items on which the task *Assess social security card (SSC) application* relies are the Shanghai SSC Policy and the Shanghai SSC Supplement where essential rules are specified for applying a SSC. An example rule states that the persons older than 18 can apply for SSCs.

Table 1. A task model

Task Model	Task Analysis work-sheet
Task	Assess social security card (SSC) application
Organization	Community Branch offices (CBOs)
Goal&Value	To make decision about the legitimacy of a SSC application
Dependency and Flow	Input task: check availability and consistency
	Output task: 1) notify citizens of results 2) take pictures
Objects Handled	Input objects: SSC application Output objects: Evaluation results
Knowledge assets	Regulations about applying for a SSC Sources: 1) Shanghai SSC Policy (S-SSC-P) 2) Shanghai SSC Supplement (S-SSC-S)
Resource	-

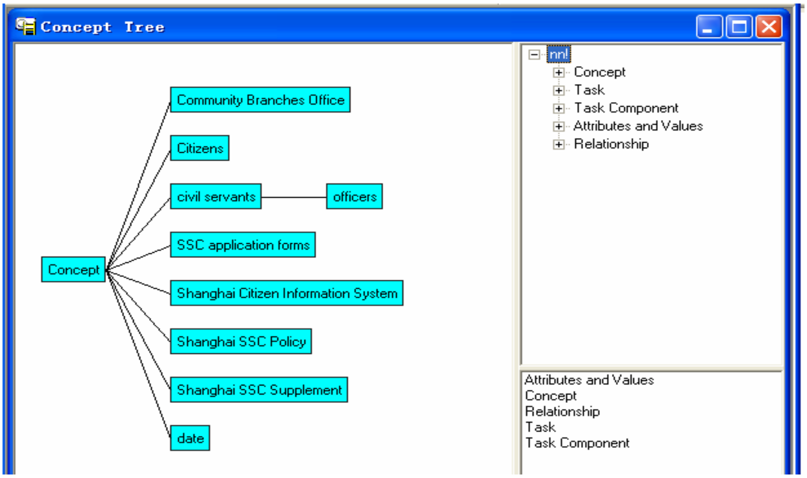


Fig. 2. PC-PACK laddering tools

4 Knowledge Acquisition

Knowledge acquisition is an effective means to extract domain concepts and domain knowledge from knowledge sources. For the SSC application, we utilize a number of various knowledge acquisition techniques such as interviews, protocol analysis, concept sorting to capture domain knowledge from experts and officers in related government agencies. The procedure of knowledge elicitation is facilitated through the use of the PC-PACK toolkit. Figure 2 shows the snapshot where the PC-PACK laddering tools is used to organize these concepts into hierarchies.

5 Knowledge Models and Ontology Modeling

Through KA techniques and tools such as PC-PACK, the CommonKADS knowledge models can be modeled. Among the knowledge models, task knowledge describes the knowledge-intensive tasks and corresponding methods that decompose these tasks into a set of sub-tasks and inference steps to solve them. To encourage the reuse of reasoning mechanisms across domains, the task and method knowledge wthin the CommonKADS are represented independently of domains. For common knowledge-intensive tasks, the CommonKADS defines corresponding task templates containing the tasks and their methods. As a result, concrete applications can directly use or customize them. For the task "*Assess social security card (SSC) application*", for instance, we can directly adopt the task template "assessment" to solve this problem.

Domain knowledge involved in knowledge models mainly defines domain concepts and relationships among them. Due to both the difficulties in reasoning consistencies caused by the non-formal semantics of CML language and the few tool supports, we adopt OWL to formally define these concepts, relationships, properties and axioms within e-government domain. A main advantage of using OWL is to enable automatic reasoning on consistencies owing to its DL-based semantics. Additionally, a range of existing tools such as Protégé, Jena can be used to manipulate OWL-based ontologies and knowledge bases.

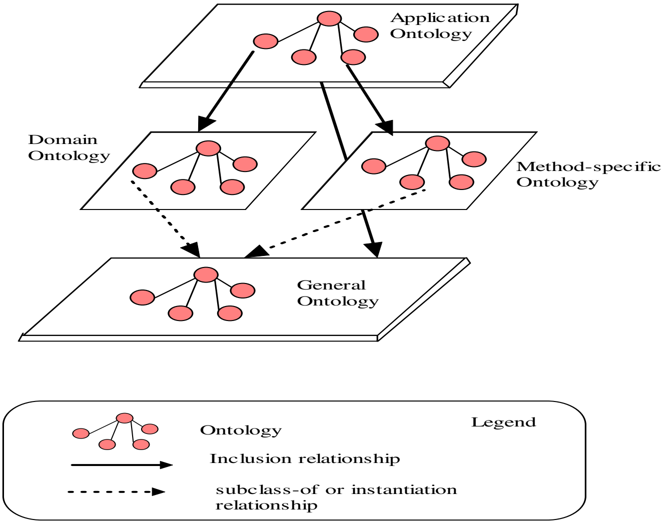


Fig. 3. Hierarchies of ontologies

To support the reuse of ontology bases, the e-government ontologies are classified into general ontology, domain ontology and application ontology, as shown in figure 3. The general ontology includes the concepts common to all domains, such as time, location and event. The domain ontology defines the concepts specific to electronic government domain, for examples, *GovDocuments*, *GovLegistilation*, *GovService*, *Citizen*. The application ontology describes the concepts and their relationships related to solving a particular application. In addition to the inclusion of the domain ontologies and general ontologies, method-specific or task-specific ontologies are contained in the application ontology. Some example application ontologies we developed in our project are *SSC–Application-Form, SSC decision*. We use the Protégé-OWL [10] to model e-government ontology, including general ontology, domain ontology and application ontology.

6 Rule Knowledge

The rule types within the commonKADS knowledge models are defined in the form of ante-consequence and consequence. Similarly to database schema, they define the schema to which rule instances conforms. Instead of using CML language, we employ SWRL [9], the rule language for semantic web, to represent the rule knowledge within e-government domain. To take an example, the rule stating that persons aged over 18 and reside in Shanghai can apply for social security cards is expressed in SWRL:

$$\begin{gathered} citizen(?x) \wedge hasage(?x, ?y) \wedge xsd: unsignedInt(?\,y) \\ \wedge swrlb: greaterThanOrEqual\ (?y, 18) \wedge\ hasResidence(?x, \mathrm{SH}) \\ \wedge\ SSCServices(?z) \rightarrow\ appliedfor(?z, ?x) \end{gathered}$$

where *citizen*, *hasage*, *hasResidence*, *SSCService*, *appliedfor* are the ontologies defined in OWL, and *Swrlb*: *greathanOrEqual* is a built-in predicate of the SWRL Specification.

7 Method Knowledge

To a knowledge-intensive task, a task method defines how to decompose the task into a set of subtasks and primitive inferences, i.e. the lowest subtask, to solve the problem. The CommonKADS identifies task templates, namely task methods, for typical knowledge-intensive tasks [4]. To take an example, for the aforementioned task "*assessment*", the corresponding method contains *abstract*, *specify*, *choose* and *evaluate* subtasks [4]. The *abstract* subtask is used to simply case data, whereas the *specify* subtask is employed to determine norms related to some case data. The *choose* subtask choose one of several norms to assess cases. The *evaluate* determine whether the cases is legitimate or not according to the results. OWL-S [8] is employed to represent the task method including the subtasks and control flow among them.

8 Conclusion

By using both the CommonKADS and the Semantic Web technologies, an approach to building an e-government KBS for automatically solving knowledge-intensive tasks is presented in this paper. Our experiences show that they can complement each other although there exists a little overlap in modeling knowledge between the CommonkADS and the semantic web technologies. Further work will be undertaken to integrate this e-government knowledge based system into a workflow management system (WMS) so that the overall government processes can be supervised and managed by the WMS.

Acknowledgments. This study is funded by Natural Science Foundation of China (NSFC) under No. 70471023.

References

1. Oreste S., Franco C., Maurizio P.: E-Government: Challenges and Opportunities, CMG Italy-XIX Annual Conference, 2005
2. Shivakumar K.: An Overview of E-Government, International Symposium on Learning Management and Technology Development in the Information and Internet Age, 2002
3. Papavassiliou, G., Ntioudis, S., Abecker, A., Mentzas, G.: Supporting Knowledge-Intensive Work in Public Administration Processes, Knowledge and Process Management, 10(3), 2003
4. Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R. Shadbold, N., van der Velde, W., Wielinda, B.: Knowledge Engineering and Management, The CommonKADS Methodology. The MIT Press, Cambrigde (2000)
5. G. Schreiber, B. Wielinga, H. Akkermans, W. Van de Verlde, and A. Anjewierden.: CML: The CommonKADS Conceptual Modeling Language, 8th European Knowledge Acquistion workshop, 1994
6. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness.: OWL Web Ontology Language Reference. 2004
7. PC PACK Knowledge tool, http://www.epistemics.co.uk/products/pcpack/
8. The OWL Services Coalition, OWL-S: Semantic Mark-up for Web Services v.1.0. Available at http://www.daml.org/services/owl-s/1.0/owl-s.pdf
9. SWRL Specification. http://www.w3.org/Submission/SWRL/
10. Protégé OWL Plugin - Ontology Editor for the Semantic Web, http://protege.stanford.edu/plugins/owl/

A Semantics-Based Protocol for Business Process Transactions

Dongwoo Kang[1], Sunjae Lee[1], Kwangsoo Kim[1], and Jae Yeol Lee[2]

[1] Dept. of Industrial and Management Engineering, Pohang University of Science and Technology, San 31, Hyoja-dong, Namgu, Pohang, Gyungbuk 790-784, South Korea
{hyunil, supersun, kskim}@postech.ac.kr
[2] Dept. of Industrial Engineering, Chonnam National University, 300 Yongbong-dong, Buk-gu, Gwangju 500-757, South Korea
jaeyeol@chonnam.ac.kr

Abstract. A Business Process Management System (BPMS) requires transaction management to guarantee reliability of the business process transactions. Several transaction protocols have been suggested for the transaction management, but they are heterogeneous. This heterogeneity interrupts message exchanges among BPMSs which use different transaction protocols, so that the interoperability among the BPMSs cannot be guaranteed. To solve this problem, this paper suggests a semantics-based protocol for business process transactions. The suggested protocol is composed of the static semantics and the operational semantics. In the context of the static semantics, transaction states and messages are defined using the Web Ontology Language (OWL). In the context of the operational semantics, state transitions of business process transactions are defined using the Abstract State Machine (ASM). The suggested approach is expected to enhance interoperability among heterogeneous BPMSs, to increase the understandability for the transaction protocols, and to support automatic transaction execution and systematic transaction monitoring.

1 Introduction

Service-oriented business processes can be organized and executed more quickly by adopting Web services as their interface. However, unexpected errors occurred in the service-oriented business processes cause fatal damages to the reliability of business processes. In order to prevent these damages, transaction management for the service-oriented business process is required.

The transaction management for the Data Base Management System (DBMS) must conform to the ACID (Atomicity, Consistency, Isolation, and Durability) property. This property originated from the DBMS-specific characteristics such as 1) tightly-coupled system, 2) short transaction processing duration, 3) strong reliability among transaction participants, and 4) transaction execution under an authorized transaction manager. On the contrary to the DBMS, the service-oriented Business Process Management System (BPMS) has the characteristics such as 1) loosely-coupled system, 2)

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 343 – 350, 2006.
© Springer-Verlag Berlin Heidelberg 2006

long transaction processing duration, 3) unreliability among transaction participants, and 4) transaction execution without an authorized transaction manager. Due to differences between the DBMS and the BPMS, it is difficult to apply the DBMS-specific transaction management method to the BPMS [6].

To solve this problem, heterogeneous transaction protocols for the BPMS have been suggested in the Web service environment. The WS-Transaction (WS-T) [2] [3], the Business Transaction Protocol (BTP) [8], and the Web Services Transaction Management (WS-TXM) [4] are representative business process transaction protocols in the Web service environment, and they use a common transaction mechanism called Two Phase Commit (2PC). However, a BPMS using one transaction protocol cannot correctly understand transaction processes, messages, and states of other BPMSs using other transaction protocols, because same concepts of the 2PC used for business process transactions are expressed differently by heterogeneous protocols.

To decrease misunderstanding caused by the heterogeneous transaction protocols, the heterogeneously-expressed transaction processes, messages and states are classified into several groups based on the semantic similarity. Based on such classification, a semantics-based protocol for business process transactions is suggested in this paper.

The suggested semantics for a transaction protocol is composed of two semantics - the static semantics and the operational semantics. In the context of the static semantics, static factors of a transaction protocol such as protocol states and protocol messages are defined as ontology using the Web Ontology Language (OWL) [10]. The static semantics conceptualizes the classified groups including transaction states and messages, and then makes relations the conceptualized groups with those of other transaction protocols. Based on such conceptualizations and relationships, one transaction protocol is expected to be able to understand corresponding transaction states and messages in other transaction protocols. Such understanding can be used to solve the problem from heterogeneous expressions for the concepts. In the context of the operational semantics, state transitions of business process transactions triggered by protocol messages are defined using the Abstract State Machine (ASM) [5]. Because the ASM is the methodology to define machine-readable state transition models based on mathematics, an ASM model of the operational semantics is expected to guarantee formalism for transaction operations and correct understanding.

This paper is organized as follows. In section 2, related works are introduced. The homogeneity of transaction mechanisms and the heterogeneity of the transaction expressions are discussed in section 3. In section 4, semantics-based transaction protocol is defined based on the static semantics and the operational semantics. Section 5 concludes this paper.

2 Related Works

As mentioned in section 1, because the characteristics of the BPMS are different from those of the DBMS, the BPMS-specific transaction protocol is required. In the Web service environment, three major BPMS-specific protocols have been suggested. First, the WS-T is composed of the WS-AtomicTransactions (WS-AT) [2] and the WS-BusinessActivity (WS-BA) [3]. The WS-AT is the transaction protocol conforming to the ACID property like the DBMS-specific protocols, and the WS-BA is the transaction protocol relaxing atomicity and isolation of the ACID property in order to execute business process transactions in the Web service environment. Second, the BTP also supports two types of transactions just as the WS-T. The *Atoms* of BTP is the transaction protocol supporting basic transactions just as the WS-AT, but it is different from the WS-AT in that it relaxes isolation of the ACID property. The *Cohesions* of the BTP is the transaction protocol supporting business process transactions in the Web service environment just as the WS-BA. Last, the WS-TXM is the other protocol composed of the TX-ACID, the TX-LongRunningAction (TX-LRA), and the TX-BusinessProcess (TX-BP), but it is not famous as above two protocols. Besides the BPMS-specific protocols, the ACTA can be used to describe the BPMS-specific protocol, because it allows the arbitrary modeling of transaction models [11].

The research on endowing transactions with semantics has received relatively little coverage in the related literature. Adams et al. study on the ontology for online service transaction [1]. However, this research provided a list of terms used in online transaction using a natural language rather than semantics of transaction. Because a natural language is not accurate as well as not machine-readable, it is necessary for transaction ontology to be modeled using ontology modeling languages such as OWL. Prinz et al. defined a DBMS-specific operational semantics using ASM, but it dose not reflect characteristics of the service-oriented business process transaction [9]. The Web Service Modeling Ontology (WSMO) [7] and OWL-S supports to model interactions, interfaces and capabilities of Web services using the ontology. Contrary to the WSMO and the OWL-S, the suggested approach supports the modeling of Web service transaction operations based on the OWL and the ASM.

3 Homogeneity of Transaction Mechanisms and Heterogeneity of Transaction Expressions Among Transaction Protocols

Although heterogeneous transaction protocols prevent BPMSs from interoperating with each other, the transaction mechanisms included in the heterogeneous transaction protocols are homogeneous. This section shows the homogeneous transaction mechanisms and the heterogeneous expressions of individual transaction protocols.

The transaction mechanisms of the WS-BA and the BTP are based on the Two-phase commit, which a coordinator and a participant execute transaction from a pre-commit phase to a commit phase. Because of this common transaction mechanism, two protocols have similar state transition diagrams. The WS-BA executes the pre-commit phase which includes "Active", "Completing" and "Completed" states, then it executes the commit phase including "Closing" and "Ended" states. Similar to the WS-BA, the BTP executes the pre-commit phase including "Enrolled", "Preparing", and "Prepared" states, then it executes the commit phase including "Confirming" and "Confirmed" states.

In spite of this common transaction mechanism, the two transaction protocols express protocol states and protocol messages heterogeneously. The table 1 shows the heterogeneous terms for similar transaction states among several transaction protocols. For example, the "Completing" and the "Completed" states in the WS-BA, which mean the preparation and approval phase of the Web service interaction closing, are expressed as "Preparing" and "Prepared" states in the BTP.

Not only the terms for transaction states but also the terms for transaction messages are expressed heterogeneously in the protocols. These heterogeneous expressions for the same states and messages cause that BPMS using one transaction protocol cannot understand transactions described by other transaction protocols or vice versa.

Before the construction of the semantics-based transaction protocol, the transaction states and messages in heterogeneous protocols should be classified into groups by business process transaction experts as shown in the Table 1.

Table 1. Heterogeneous expressions of terms for the transaction states in protocols

Phase		BTP	WS-AT	WS-BA(PC)	WS-BA(CC)	WS-TXM (ACID)
Pre-commit Phase	Register	Enrolling	-	-	-	-
		Enrolled	Active	Active	Active	Active
	Prepare	Preparing	Preparing	-	Completing	Preparing
		Prepared	Prepared	Completed	Completed	Prepared
	Resign	Resigning	-	Exiting	Exiting	-
		Resigned	-	Ended	Ended	-
Commit phase	Commit	One-phase-confirming	-	-	-	One-phase-commit
		Confirming	Committing	Closing	Closing	Committing
		Confirmed	Ended	Ended	Ended	Committed
	Cancel	Canceling	Aborting	Canceling Compensating	Canceling Compensating	RollingBack
		Cancelled	Ended	Ended	Ended	RolledBack
	Error	Contradicting	-	Faulting	Faulting	-
		Contradiction	-	Ended	Ended	-

4 Semantics-Based Transaction Protocol

The semantics-based transaction protocol plays the role of a bridge among heterogeneous transaction protocols. If the semantics-based transaction protocol is defined, concepts of existing transaction protocols would be mapped to the semantics-based protocol using semantic matching. Such mapping is expected to increase understandability of the existing transaction protocols.

4.1 Formalization of the Static Semantics Using the OWL

The static semantics can be defined as ontologies which describe protocol states and messages. To define the ontologies, protocol states and messages of the transaction protocols must be classified by the similarity of meanings as shown in Table 1. The classified concepts for the states and messages are formalized as ontologies, and relations between the concepts are established. The static semantics is modeled using the ontology modeling language, the OWL, as shown in Fig. 1. The static semantics defined in this manner can support interoperability among transaction protocols by

```
  <?xml version="1.0"?>
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://www.TPO.com/ProtocolRole"/>
    <owl:imports rdf:resource="http://www.TPO.com/ProtocolState"/>
  </owl:Ontology>
  <owl:Class rdf:ID="ENROL">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">A request to
    a Coordinator to enrol a Participant</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Message"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Restriction>
    <owl:ObjectProperty rdf:ID="participant">
      <rdfs:domain rdf:resource="#ENROL"/>
      <rdfs:range rdf:resource="http://www.TPO.com/ProtocolRole#Participant"/>
  </owl:ObjectProperty>
    <owl:cardinality>1</owl:cardinality>
  </owl:Restriction>
  <owl:DatatypeProperty rdf:ID="expireDate">
    <rdfs:domain rdf:resource="#ENROL"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#time"/>
  </owl:DatatypeProperty>
  ...
```

Fig. 1. Formalization of the "ENROL" message using the OWL

```
<owl:Class rdf:ID="REGIST">
 <owl:sameAs rdf:resource="http://www.TPO.com/ProtocolMsg#ENROL"/>
</owl:Class>
```

Fig. 2. Case that the "ENROL" and the "REGIST" are same

```
<owl:Class rdf:ID="ENROL">
 <owl:differentFrom rdf:resource="http://www.TPO.com/ProtocolMsg#ENROL"/>
</owl:Class>
```

Fig. 3. Case that two "ENROL"s are different from each other

forming relations with each other. Such relations can be modeled with OWL tags, ***owl:sameAs***, ***owl:differentFrom***, and so on. Fig.2 and Fig.3 show that two concepts are same and different, respectively.

Based on this conceptualizations and relationships of the static semantics, one transaction protocol is expected to be able to understand corresponding transaction states and messages in other transaction protocols. As the static semantics for transaction protocols are defined in this manner, the protocols are expected to be machine-readable and the problem from heterogeneous expressions for states and messages among various transaction protocols can be solved.

4.2 Formalization of the Operational Semantics Using the ASM

The operational semantics can be defined by modeling state transitions of business process transactions using the ASM. The ASM is the methodology that models state machines based on the mathematical formalism. In this paper, Abstract state machine Language (AsmL), one of the machine-readable ASM modeling language, is used to formalize the operational semantics. The ASM expresses state transitions using an "if..then..else.." sentence, and Fig. 4 shows a part of the ASM model of the operational semantics for the semantics-based transaction protocol using the AsmL.

```
enum EnumState
  Enrolling:uri="http://www.TPO.com/ProtocolState#Enrolling"

  Enrolled:uri="http://www.TPO.com/ProtocolState#Enrolled"

   ...
enum EnumMessages
 ENROL:uri="http://www.TPO.com/ProtocolMessage#ENROL"

 ENROLLED:uri="http://www.TPO.com/ProtocolMessage#ENROLLED"

   ...
Main()
  initially currentTransactionState as EnumState = BeforTransaction
  initially inputMessage = any message | message in {ENROL, ENROLLED, ...... }
   step until currentTransactionState = Cancelled or currentTransactionState = Resigned
or currentTransactionState = Contradiction or currentTransactionState = Confirmed
    //Transaction when Enrolling state
    if currentTransactionState = Enrolling then
      if inputMessage = ENROLLED then
        step currentTransactionState := Enrolled
    //Transaction when Enrolled state
    if currentTransactionState = Enrolled then
      if inputMessage = ONE_PHASE_CONFIRM then
        step currentTransactionState := One_phase_confirming
          ...
    temp = any message | message in {ENROL, ENROLLED, PREPARE, ......}
    inputMessage :=temp
```

Fig. 4. A part of an ASM model for the operational semantics of the semantics-based protocol

In order for the operational semantics to use the pre-defined states and messages of the static semantics, the Universal Resource Identifier (URI) can be used. However, because the AsmL, the modeling language for the operational semantics, does not support the URI, an "***uri***" directive is suggested in this paper. The sentence with the "***uri***" directive means that the variable defined with the "***uri***" uses its value which has a type of the URI and indicates the state and message ontology in the static semantics.

Because the ASM is the methodology to define machine-readable state transition models based on mathematics, an ASM model representing the operational semantics can guarantee formalism for transaction operations, and it can be utilized for operations of business process transactions to be understood easily by BPMSs. In addition, machine-readability of the ASM model supports automatic business process transactions in the phase of transaction execution and transaction state transition described in the ASM model supports systematic monitoring for the state of the business process transaction.

5 Conclusion and Discussion

Heterogeneous transaction protocols for the business process transaction management have been suggested, but their heterogeneous expressions are an obstacle to guaranteeing interoperability among the BPMSs. In order to solve such problem, a semantics-based protocol for business process transactions is suggested in this paper. The suggested protocol is composed of two semantics - the static semantics and the operational semantics. In the context of the static semantics, transaction states and messages are defined using the OWL, and in the context of the operational semantics, state transitions of business process transactions are defined using the ASM. Based on the semantics, the semantics-based neutral transaction protocol which plays the role of a bridge among transaction protocols can be defined. Consequently, the suggested approach is expected to enhance interoperability among BPMSs using heterogeneous transaction protocols, to increase the understandability for the transaction protocols, and to support automatic transaction execution and systematic transaction monitoring.

Acknowledgement

This work is supported by grant No. R01-2003-000-10171-0 from the Basic Research Program of the Korea Science and Engineering Foundation.

References

1. Adams, N., Fraser, J., Macintosh, A., and McKay-Hubbard, A.: Towards an Ontology for Electronic Transaction Services. Int. J. Intell. Sys. Acc. Fin.Mgmt. 11(2002) 173–181
2. Arjuna, BEA, Hitachi, IBM, IONA, and Microsoft: Web Services Atomic Transaction (WS-AtomicTransaction) specification. (2005) ftp://www6.software.ibm.com/software/developer/library/WS-AtomicTransaction.pdf
3. Arjuna, BEA, Hitachi, IBM, IONA, and Microsoft: Web Services Business Activity Framework (WS-BusinessActivity) specification. (2005) ftp://www6.software.ibm.com/software/developer/library/WS-BusinessActivity.pdf
4. Arjuna, Fujitsu, IONA, Oracle, and Sun Microsystems: Web Services Transaction Management (WS-TXM) specification. (2003) http://developers.sun.com/techtopics/webservices/wscaf/wstxm.pdf

5. Börger, E. and Stärk, R.: Abstract State Machines: A Method for High-Level System Design and Analysis. Springer-Verlag, USA (2003)
6. Dalal, S. Little, M. Potts, M. Temel, S., and Webber, J.: Coordinating Business Transactions on the Web. IEEE Internet Computing Special Edition on Web Services(2003) 30-39
7. ESSI WSMO Working group: Web Service Modeling Ontology (WSMO) specification. (2005) http://www.wsmo.org/TR/d2/v1.2/
8. OASIS: Business Transaction Protocol. (2004) http://xml.coverpages.org/BTPv11-200411.pdf
9. Prinz, A. and Thalheim, B.: Operational Semantics of Transactions. Proceedings of the Fourteenth Australasian database conference on Database technologies 17(2003) 169-179
10. W3C: OWL Web Ontology Language Overview. (2004) http://www.w3.org/TR/owl-features/
11. Chrysanthis, P. K., and Ramamritham, K., Synthesis of Extended Transaction Models Using ACTA, ACM Transactions on Database Systems, 19:3 (1994) 450-491

Fuzzy View-Based Semantic Search

Markus Holi and Eero Hyvönen

Helsinki University of Technology (TKK), Media Technology and University of Helsinki
P.O. Box 5500, FI-02015 TKK, Finland
firstname.lastname@tkk.fi
http://www.seco.tkk.fi/

Abstract. This paper presents a fuzzy version of the semantic view-based search paradigm. Our framework contributes to previous work in two ways: First, the fuzzification introduces the notion of relevance to view-based search by enabling the ranking of search results. Second, the framework makes it possible to separate the end-user's views from content indexer's taxonomies or ontologies. In this way, search queries can be formulated and results organized using intuitive categories that are different from the semantically complicated indexing concepts. The fuzziness is the result of allowing more accurate weighted annotations and fuzzy mappings between search categories and annotation ontologies. A prototype implementation of the framework is presented and its application to a data set in a semantic eHealth portal discussed.

1 Introduction

Much of semantic web content will be published using semantic portals[1] [16]. Such portals usually provide the user with two basic services: 1) A search engine based on the semantics of the content [6], and 2) dynamic linking between pages based on the semantic relations in the underlying knowledge base [9]. In this paper we concentrate on the first service, the semantic search engine.

1.1 View-Based Semantic Search

The view-based search paradigm[2] [23,11,13] is based on *facet analysis* [18], a classification scheme introduced in information sciences by S. R. Ranganathan already in the 1930's. From the 1970's on, facet analysis has been applied in information retrieval research, too, as a basis for search. The idea of the scheme is to analyze and index search items along multiple orthogonal taxonomies that are called subject *facets* or *views*. Subject headings can then be synthesized based on the analysis. This is more flexible than the traditional library classification approach of using a monolithic subject heading taxonomy.

In view-based search, the views are exposed to the end-user in order to provide her with the right query vocabulary, and for presenting the repository contents and search results along different views. The query is formulated by constraining the result set in the

[1] See, e.g., http://www.ontoweb.org/ or http://www.semanticweb.org

[2] A short history of the parading is presented in http://www.view-based-systems.com/history.asp

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 351–365, 2006.
© Springer-Verlag Berlin Heidelberg 2006

following way: When the user selects a category c_1 in a view v_1, the system constrains the search by leaving in the result set only such objects that are annotated (indexed) in view v_1 with c_1 or some sub-category of it. When an additional selection for a category c_2 from another view v_2 is made, the result is the intersection of the items in the selected categories, i.e., $c_1 \cap c_2$. After the result set is calculated, it can be presented to the end-user according to the view hierarchies for better readability. This is in contrast with traditional search where results are typically presented as a list of decreasing relevance.

View-based search has been integrated with the notion of ontologies and the semantic web [13,21,12,17]. The idea of such *semantic view-based search* is to construct facets algorithmically from a set of underlying ontologies that are used as the basis for annotating search items. Furthermore, the mapping of search items onto search facets could be defined using logic rules. This facilitated more intelligent "semantic" search of indirectly related items. Another benefit is that the logic layer of rules made it possible to use the same search engine for content annotated using different annotation schemes. Ontologies and logic also facilitates *semantic browsing*, i.e., linking of search items in a meaningful way to other content not necessarily present in the search set.

1.2 Problems of View-Based Search

View-based search helps the user in formulating the query in a natural way, and in presenting the results along the views. The scheme has also some shortcomings. In this paper we consider two of them:

Representing relevance. View-based search does not incorporate the notion of relevance. In view-based search, search items are either annotated using the categories or mapped on them using logic rules. In both cases, the search result for a category selection is the crisp set of search items annotated to it or its sub-concepts. There is no way to rank the results according to their relevance as in traditional search. For example, consider two health-related documents annotated with the category Helsinki. One of the documents could describe the health services in Helsinki, the other could be a European study about alcohol withdrawal syndromes of heavy alcohol users, for which the research subject were taken randomly from London, Paris, Berlin, Warsaw and Helsinki. It is likely that the first document is much more relevant for a person interested in health and Helsinki.

Separating end-user's views from indexing schemes. Annotation concepts used in annotation taxonomies or ontologies often consist of complicated professional concepts needed for accurate indexing. When using ontologies, the annotation concepts are often organized according a formal division of the topics or based on an upper-ontology. This is important because it enables automatic reasoning over the ontologies. However, such categorizations are not necessarily useful as search views because they can be difficult to understand and too detailed from the human end-users viewpoint. The user then needs a view to the content that is different from the machine's or indexer's viewpoint. However, current view-based system do not differentiate between indexer's, machine's, and end-user's views. In our case study, for example, we deal with problem of publishing health content to ordinary citizens in a coming semantic portal *Tervesuomi.fi*. Much of the material to be used has been indexed using complicated medical terms and classifications, such as Medical Subject Headings[3] (MeSH). Since the end-user is not an expert of the domain and is not familiar with the professional terms used in the ontology, their hierarchical organization is not suitable for formulating end-user queries or presenting the result set, but only for indexing and machine processing.

This paper presents a fuzzy version of the semantic view-based search paradigm in which 1) the degrees of relevance of documents can be determined and 2) distinct end-user's views to search items can be created and mapped onto indexing ontologies and the underlying search items (documents). The framework generalizes view-based search from using crisp sets to fuzzy set theory and is called *fuzzy view-based semantic search*. In the following, this scheme is first developed using examples from the *Tervesuomi.fi* portal content. After this an implementation of the system is presented. In conclusion, contributions of the work are summarized, related work discussed, and directions for further research proposed.

2 Fuzzy View-Based Semantic Search

2.1 Architecture of the Framework

Figure 1 depicts the architecture of the fuzzy view-based semantic search framework. The framework consists of the following components:

Search Items. The search items are a finite set of documents D depicted on the left. D is the fundamental set of the fuzzy view-based search framework.

Annotation Ontology. The search items are annotated according to the ontology by the indexer. The ontology consists of two parts. First, a finite set of annotation concepts AC, i.e. a set of fuzzy subsets of D. Annotation concepts $AC_i \in AC$ are atomic. Second, a finite set of annotation concept inclusion axioms $AC_i \subseteq AC_j$[4], where $AC_i, AC_j \in AC$ are annotation concepts and $i, j \in N$, and $i \neq j$. These inclusion axioms denote subsumption between the concepts and they constitute a concept hierarchy.

Search Views. Search views are hierarchically organized search categories for the end-user to use during searching. The views are created and organized with end-user interaction in mind and may not be identical to the annotation concepts. Each search category SC_i is a fuzzy subset of D. In crisp view-based search the intersection of documents related to selected search categories is returned as the result set, while in fuzzy view-based search, the intersection is replaced by the fuzzy intersection.

Search items related to a search category SC_i can be found by mapping them first onto annotation concepts by annotations, and then by mapping annotation concepts to SC_i. The result R is not a crisp set of search items $R = SC_1 \cap ... \cap SC_n =$

[3] http://www.nlm.nih.gov/mesh/

[4] Subset relation between fuzzy sets is defined as: $AC_i \subseteq AC_j$ iff $\mu_{AC_j}(D_i) \geq \mu_{AC_i}(D_i)$, $\forall D_i \in D$, where D is the fundamental set.

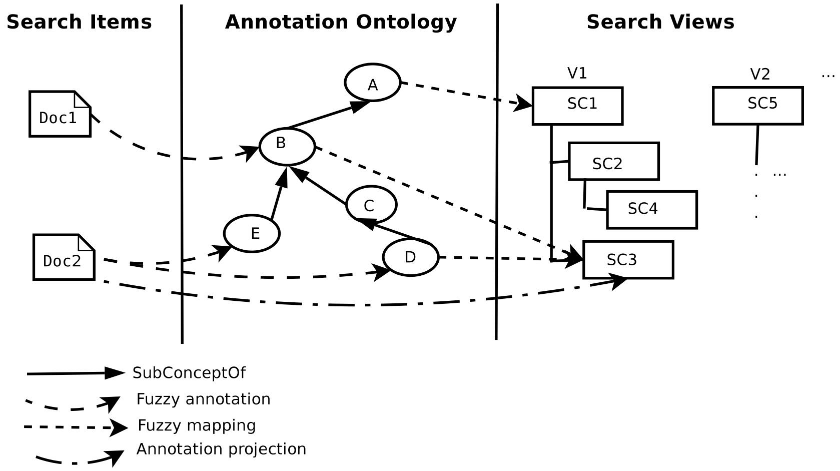


Fig. 1. Components of fuzzy view-based semantic search framework

$\{Doc_1, ..., Doc_m\}$ as in view-based search, but a fuzzy set where the relevance of each item is specified by the value of the membership function mapping:

$R = SC_1 \cap ... \cap SC_n = \{(Doc_1, \mu_1), ..., (Doc_, \mu_m)\}$.

In the following the required mappings are described in detail.

2.2 Fuzzy Annotations

Search items (documents) have to be annotated in terms of the annotation concepts—either manually or automatically by using e.g. logic rules. In (semantic) view-based search, the annotation of a search item is the crisp set of annotation concept categories in which the item belongs to. In figure 1, annotations are represented using bending dashed arcs from *Search Items* to *Annotation Ontology*. For example, the annotation of item *Doc2* would be the set $A_{Doc2} = \{E, D\}$.

In our approach, the relevance of different annotation concepts with respect to a document may vary and is represented by a *fuzzy annotation*. The fuzzy annotation A_D of a document D is the set of its fuzzy concept membership assertions:

$A_D = \{(AC_1, \mu_1), ..., (AC_n, \mu_n)\}$, where $\mu_i \in (0, 1]$.

Here μ_i tell the degrees by which the annotated document is related to annotation concepts AC_i. For example;

$A_{D1} = \{(Exercise, 0.3), (Diet, 0.4)\}$

Based on the annotations, the membership function of each fuzzy set $AC_j \in AC$ can be defined. This is done based on the meaning of subsumption, i.e. inclusion. One concept is subsumed by the other if and only if all individuals in the set denoting the subconcept are also in the set denoting the superconcept, i.e., if being in the subconcept implies being in the superconcept [24]. In terms of fuzzy sets this means that $AC_i \subseteq AC_j$, and $\mu_{AC_i}(D_i) = \nu$ implies that $\mu_{AC_j}(D_i) \geq \nu$, where $\nu \in (0, 1]$, and D_i is a search item, and $\mu_{AC_i}(D_i)$, and $\mu_{AC_j}(D_i)$ are the membership functions of sets AC_i and AC_j respectively.

Thus, we define the membership degree of a document D_i in AC_j as the maximum of its concept membership assertions made for the subconcepts of AC_j.

$\forall D_i \in D, \mu_{AC_j}(D_i) = max(\mu_{AC_i}(D_i))$, where $AC_i \subseteq AC_j$.

For example, assume that we have a document $D1$ that is annotated with annotation concept *Asthma* with weight 0.8, i.e. $\mu_{Asthma}(D1) = 0.8$. Assume further, that in the annotation ontology *Asthma* is a subconcept of *Diseases*, i.e. $Asthma \subseteq Diseases$. Then,

$\mu_{Diseases}(D1) = \mu_{Asthma}(D1) = 0.8$.

2.3 Fuzzy Mappings

Each search category SC_i in a view V_j is defined using concepts from the annotation ontology by a finite set of fuzzy concept inclusion axioms that we call *fuzzy mappings*:

$AC_i \subseteq_\mu SC_j$, where $AC_i \in AC, SC_j \in V_k, i, j, k \in N$ and $\mu \in (0, 1]$

A fuzzy mapping constrains the meaning of a search category SC_j by telling to what degree μ the membership of a document D_i in an annotation concept AC_i implies its membership in SC_j.

Thus, fuzzy inclusion is interpreted as fuzzy implication. The definition is based on the connection between inclusion and implication described previously. This is extended to fuzzy inclusion as in [27,5]. We use Goguen's fuzzy implication, i.e.

$i(\mu_{AC_j}(D_i), \mu_{SC_i}(D_i)) = 1$ if $\mu_{SC_i}(D_i) \geq \mu_{AC_j}(D_i)$, and $\mu_{SC_i}(D_i)/\mu_{AC_j}(D_i)$ otherwise, $\forall D_i \in D$.

A fuzzy mapping $M_k = AC_i \subseteq_\nu SC_j$ defines a set MS_k, s.t. $\mu_{MS_k}(D_l) = \nu * \mu_{AC_i}(D_l), \forall D_l \in D$, where $i(\mu_{AC_i}(D_l), \mu_{SC_j}(D_l)) = \nu$ and $\nu \in (0, 1]$. Goguen's implication was chosen, because it provides a straight-forward formula to compute the above set.

A search category SC_j is the union of its subcategories and the sets defined by the fuzzy mappings pointing to it. Using Gödel's union function[5] the membership function of SC_j is

$$\mu_{SC_j}(D_i) = max(\mu_{SC_1}(D_i), ..., \mu_{SC_n}(D_i), \mu_{MS_1}(D_i), ..., \mu_{MS_n}(D_i)), \forall D_i \in D,$$

where $SC_{1,...,n}$ are subcategories of SC_j, and $MS_{1,...,n}$ are the sets defined by the fuzzy mappings pointing to SC_j. This extends the idea of view-based search, where view categories correspond directly to annotation concepts.

Continuing with the example case in the end of section 2.2 where we defined the membership of document D_1 in the annotation concept $Diseases$. If we have a fuzzy mapping

$Diseases \subseteq_{0.1} Food\&Diseases$

[5] $\mu_{A \cup B}(D_i) = max(\mu_A(D_i), \mu_B(D_i)), \forall D_i \in D$.

then the membership degree of the document $D1$ in $Food\&Diseases$ is

$$\mu_{Food\&Diseases}(D1) = \mu_{Diseases}(D1) * 0.1 = 0.8 * 0.1 = 0.08.$$

Intuitively, the fuzzy mapping reveals to which degree the annotation concept can be considered a subconcept of the search category. Fuzzy mappings can be created by a human expert or by an automatic or a semi-automatic ontology mapping tool. In figure 1, fuzzy mappings are represented using straight dashed arcs.

The fuzzy mappings of a search category can be *nested*. Two fuzzy mappings $M_1 = AC_i \subseteq_\mu SC_i$ and $M_2 = AC_j \subseteq_\nu SC_i$ are *nested* if $AC_i \subseteq AC_j$, i.e., if they point to the same search category, and one of the involved annotation concepts is the subconcept of the other. Nesting between the fuzzy mappings M_1 and M_2 is interpreted as a shorthand for $M_1 = AC_i \subseteq_\mu SC_i$ and $M_2 = (AC_j \cap \neg AC_i) \subseteq_\nu SC_i$. This interpretation actually dissolves the nesting. For example, if we have mappings

$M_1 =$ *Animal nutrition* $\subseteq_{0.1} Nutrition_{sc}$ and $M_2 =$ *Nutrition* $\subseteq_{0.9} Nutrition_{sc}$, and in the annotation ontology *Animal nutrition* $\subseteq$ *Nutrition*, then M_1 is actually interpreted as

$$M_1 = \textit{Nutrition} \cap \neg Animalnutrition \subseteq_{0.9} Nutrition_{sc}.$$

In some situations it is useful to be able to map a search category to a Boolean combination of annotation concepts. For example, if a search view contains the search category $Food\&Exercise$ then those documents that talk about both nutrition and exercise are relevant. Thus, it would be valuable to map $Food\&Exercise$ to the intersection of the annotation concepts $Nutrition$ and $Exercise$. To enable mappings of this kind, a Boolean combination of annotation concepts can be used in a fuzzy mapping. The Boolean combinations are $AC_1 \cap ... \cap AC_n$ (intersection), $AC_1 \cup ... \cup AC_n$ (union) or $\neg AC_1$ (negation), where $AC_1, ..., AC_n \in AC$.

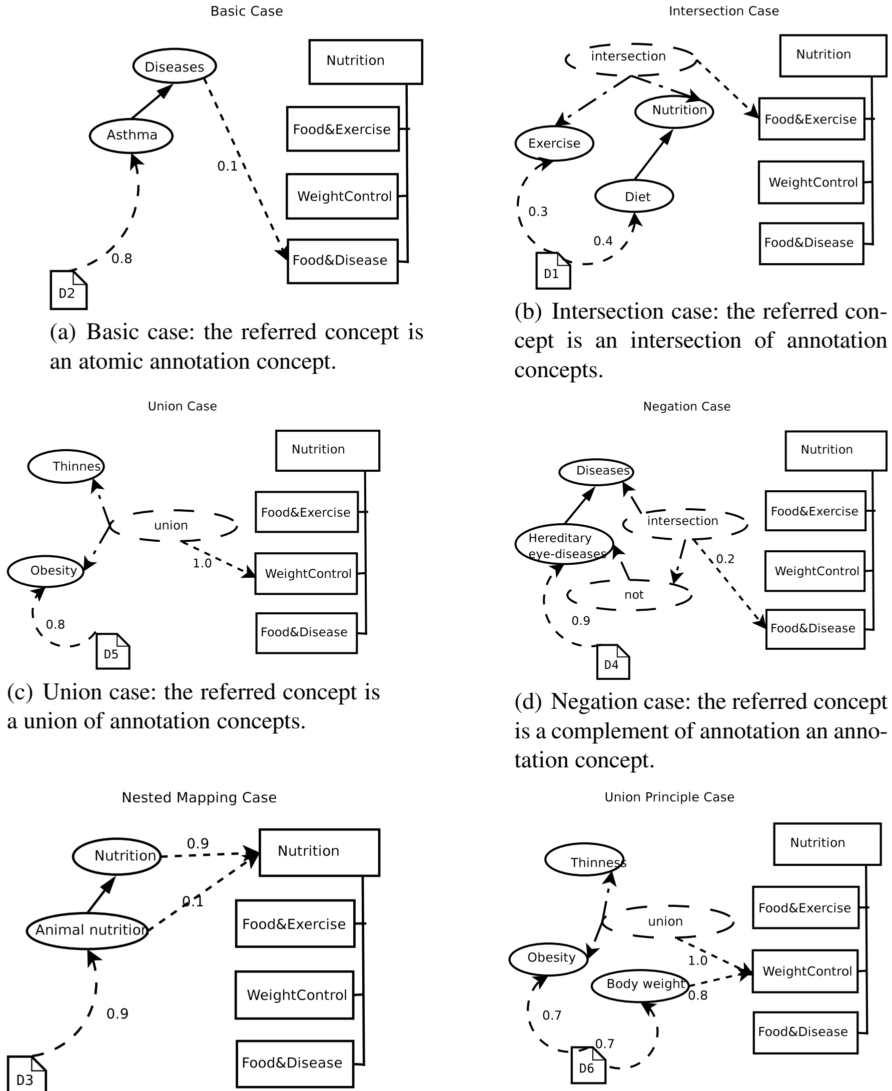
In the following, a detailed description is presented on how to determine the fuzzy sets corresponding to search categories in each of the Boolean cases. The real-world cases of figure 2 will used as examples in the text. In section 2.5 we describe how to execute the view-based search based on the projected annotations and end-user's selections.

2.4 Mappings to Boolean Concepts

In the following, the membership function definition for each type of Boolean concept is listed, according to the widely used Gödel's functions[6]:

Union Case. $AC_j = AC_k \cup ... \cup AC_n$: The membership degree of a document in AC_j is the maximum of its concept membership values in any of the components of the union concept:
$\forall D_k \in D, \mu_{AC_j}(D_k) = max(\mu_{AC_i}(D_k))$, where $i \in k, ..., n$
In the example union case of figure 2(c) we get
$\mu_{\textit{Thinnes} \cup \textit{Obesity}}(D5) = max(\mu_{\textit{Thinnes}}(D5), \mu_{\textit{Obesity}}(D5))$
$= max(0, 0.8) = 0.8.$

[6] If A and B are fuzzy sets of the fundamental set X, then $\mu_{A \cup B}(x) = max(\mu_A(x), \mu_B(x))$, $\mu_{A \cap B}(x) = min(\mu_A(x), \mu_B(x))$, and $\mu_{\neg A}(x) = 0$, if $\mu_A(x) \geq 0$, 0 otherwise, $\forall x \in X$.



(a) Basic case: the referred concept is an atomic annotation concept.

(b) Intersection case: the referred concept is an intersection of annotation concepts.

(c) Union case: the referred concept is a union of annotation concepts.

(d) Negation case: the referred concept is a complement of annotation an annotation concept.

(e) Nested mappings case: two fuzzy mappings are nested.

(f) Union Principle case: the definition of a search category is the union of its fuzzy mappings.

Fig. 2. Real-world examples of annotation projection cases

Intersection Case. $AC_j = AC_k \cap ... \cap AC_n$: The membership degree of a document in AC_j is the minimum of its concept membership values in any of the components of the union concept. $\forall D_k \in D, \mu_{AC_j}(D_k) = min(\mu_{AC_i}(D_k))$, where $i \in k, ..., n$. In the example intersection case of figure 2(b) we get

$$\mu_{Nutrition \cap Exercise}(D1) = min(\mu_{Nutrition}(D1), \mu_{Exercise}(D1))$$
$$= min(0.4, 0.3) = 0.3.$$

Negation Case. $AC_j = \neg AC_k$: The membership degree of a document in AC_j is 1 if the membership degree of the document in AC_k is 0, and 0 otherwise. $\forall D_k \in D, \mu_{AC_j}(D_k) = 0$ if $\mu_{AC_k}(D_k) > 0$ and $\mu_{AC_j}(D_k) = 1$ if $\mu_{AC_k}(D_k) = 0$. In the example negation case of figure 2(d) we get

$$\mu_{\neg \textit{Congenital diseases}}(D4) = 0 \text{ because } (\mu_{\textit{Congenital diseases}}(D4) = 0.9) > 0.$$

After the membership function of each boolean concept is defined, the membership function of the search concept can be computed based on the fuzzy mappings. For example, in figure 2(f) the projection of document $D6$ to the search view is done in the following way: The membership degrees of $D6$ in the relevant annotation concepts are

$\mu_{\textit{Thinness} \cup \textit{Obesity}}(D6) = 0.7$ and $\mu_{\textit{Body weight}}(D6) = 0.7$.
Now, the first fuzzy mapping of these yields
$\mu_{MS_1}(D6) = 0.7$
and the second one
$\mu_{MS_2}(D6) = 0.7 * 0.8 = 0.56$.

Because each search category is the union of its subcategories and the sets defined by the fuzzy mappings pointing to it, and $WeightControl$ does not have any subcategories, we get
$\mu_{WeightControl}(D6) = max(\mu_{MS_1}(D6), \mu_{MS_2}(D6)) = 0.7$.

2.5 Performing the Search

In view-based search the user can query by choosing concepts from the views. In crisp semantic view-based search, the extension E of a search category is the union of its projection P and the extensions of its subcategories S_i, i.e. $E = P \bigcup S_i$. The result set R to the query is simply the intersection of the extensions of the selected search categories $R = \bigcap E_i$ [12].

In fuzzy view-based search we extend the crisp union and intersection operations to fuzzy intersection and fuzzy union. Recall, from section 2.3 that a search category was defined as the union of its subcategories and the sets defined by the fuzzy mappings pointing to it. Thus, the fuzzy union part of the view-based search is already taken care of. Now, if E is the set of selected search categories, then the fuzzy result set R is the fuzzy intersection of the members of E, i.e. $R = SC_1 \cap ... \cap SC_n$, where $SC_i \in E$.

Using Gödel's intersection [32], we have:

$$\mu_R(D_k) = min(\mu_{SC_1}(D_k), ..., \mu_{SC_n}(D_k)), \forall D_k \in D.$$

As a result, the answer set R can be sorted according to relevance in a well-defined manner, based on the values of the membership function.

3 Implementation

In the following an implementation of our framework is presented. In sections 3.1 and 3.2, RDF [1] representations of fuzzy annotations and search views are described, respectively. Section 3.3 presents an algorithm for the annotation projection discussed in section 2.4. Section 3.4 describes the dataset that we used to test the framework, and finally, in section 3.5 preliminary user evaluation of our test implementation is presented.

3.1 Representing Fuzzy Annotations

We created an RDF representation for fuzzy annotations. In the representation each document is a resource represented by an URI, which is the URL of the document. The fuzzy annotations of the document is represented as an instance of a ’Descriptor’ class with two properties. 1) A ’describes’ property points to a document URI, and 2) a ’hasElement’ property points to a list representing the fuzzy annotations. The fuzzy annotation is an instance of a ’DescriptorElement’ class. This class has two properties: 1) ’hasConcept’ which points to the annotation concept, and 2) ’hasWeight’, which tells the weight, i.e. the fuzziness of the annotation. For example, the fuzzy annotation of the document $D1$ in figure 2 is represented in the following way.

```
<DescriptorElement rdf:ID="descriptorelement_63">
     <hasTerm rdf:resource="&mesh;D004032"/>
     <hasWeight>0.4</hasWeight>
</DescriptorElement>
<DescriptorElement rdf:ID="descriptorelement_64">
     <hasTerm rdf:resource="&mesh;D015444"/>
     <hasWeight>0.3</hasWeight>
</DescriptorElement>
<Descriptor rdf:ID="Descriptor_6">
     <describes rdf:resource="#D1"/>
     <hasElement rdf:parseType="Collection">
          <DescriptorElement rdf:about="#descriptorelement_63"/>
          <DescriptorElement rdf:about="#descriptorelement_64"/>
     </hasElement>
</Descriptor>
```

Also the projected annotations are represented in the same manner.

Our model does not make any commitments about the method by which these fuzzy annotations are created.

3.2 Representing Search Views

We created an RDF representation of the views and the mappings between the search categories of the views and the annotation concepts. Our representation is based on the Simple Knowledge Organization System (SKOS) [3,2]. For example the search categories $Nutrition$ and $Nutrition\&Diseases$ in figure 2 are represented in the following way:

```
<skos:Concept rdf:ID="Nutrition">
     <skos:prefLabel xml:lang="en">Nutrition
     </skos:prefLabel>
     <fuzzy:mapping>
       <rdf:Description>
        <skosMap:narrowMatch rdf:resource="&mesh;D009747"/>
        <fuzzy:degree>0.9</fuzzy:degree>
       <rdf:Description>
     </fuzzy:mapping>
     <fuzzy:mapping>
       <rdf:Description>
        <skosMap:narrowMatch rdf:resource="&mesh;D000824"/>
        <fuzzy:degree>0.1</fuzzy:degree>
       <rdf:Description>
     </fuzzy:mapping>
</skos:Concept>
<skos:Concept rdf:ID="FoodAndDisease">
     <skos:prefLabel xml:lang="en">Food and Disease
```

```
      </skos:prefLabel>
      <skos:broader rdf:resource="#Nutrition"/>
      <fuzzy:mapping>
        <rdf:Description>
         <skosMap:narrowMatch>
          <skosMap:AND>
                <rdf:li rdf:resource="&mesh;Diseases"/>
                <rdf:li>
                    <skosMap:NOT>
                        <rdf:li rdf:resource="&mesh;D015785"/>
                    </skosMap:NOT>
                </rdf:li>
            </skosMap:AND>
         </skosMap:narrowMatch>
         <fuzzy:degree>0.25</fuzzy:degree>
        <rdf:Description>
      </fuzzy:mapping>
</skos:Concept>
```

We use the $narrowMatch$ property of SKOS for the mapping because it's semantics corresponds closely to the implication operator as we want: If a document d is annotated with an annotation concept AC_1, and AC_1 is a $narrowMatch$ of a search category SC_1, then the annotation can be projected from AC_1 to SC_1. The $degree$ property corresponds to the degree of truth of the mapping used in SKOS.

Our model does not make any commitments about the method by which these fuzzy mappings are created.

3.3 Projection of Annotations

We implemented the projection of annotations — i.e. the computation of the membership degrees of the documents in each search category — using the Jena Semantic Web Framework[7]. The implementation performs the following steps:

1. The RDF data described above is read and a model based on it is created. This involves also the construction of the concept hierarchies based on the RDF files.
2. The nested mappings are dissolved. This is done by running through the mappings that point to each search category, detecting the nested mappings using the concept hierarchy and dissolving the nesting according to the method described in section 2.3.
3. The membership function of each annotation concept is computed using the method described in section 2.2.
4. The membership function of each search category is computed using the method described in section 2.3.

3.4 Dataset and Ontology

Our document set consisted of 163 documents from the web site of the National Public Health Institute[8] of Finland (NPHI).

As an annotation ontology we created a SKOS translation of FinMeSH, the Finnish translation of MeSH. The fuzzy annotations were created in two steps. First, an information scientist working for the NPHI annotated each document with a number of

[7] http://jena.sourceforge.net/

[8] See, http://www.ktl.fi/

FinMeSH concepts. These annotations were crisp. Second, the crisp annotations were weighted using an ontological version of the TF-IDF [25] weighting method widely used in IR systems. We scanned through each document and weighted the annotations based on the occurrences of the annotation concept labels (including subconcept labels) in the documents. The weight was then normalized, to conform to the fuzzy set representation.

The search views with the mappings were designed and created by hand.

3.5 Evaluation

The main practical contribution of our framework in comparison to crisp view-based search is the ranking of search results according to relevance. A preliminary user-test was conducted to evaluate the ranking done by the implementation described above. The test group consisted of five subjects.

The test data was created in the following way. Five search categories were chosen randomly. These categories were: Diabetes, Food, Food Related Diseases, Food Related Allergies, and Weight Control. The document set of each category was divided into two parts. The first part consisted of the documents who's rank was equal or better than the median rank, and the second part consisted of documents below the median rank. Then a document was chosen from each part randomly. Thus, each of the chosen categories was attached with two documents, one representing a well ranking document, and the other representing a poorly ranking document.

The test users were asked to read the two documents attached to a search category, e.g. Diabetes, in a random order, and pick the one that they thought was more relevant to the search category. This was repeated for all the selected search categories. Thus, each tested person read 10 documents.

The relevance assessment of the test subjects were compared to the ordering done by our implementation. According to the results every test subject ordered the documents in the same way that the algorithm did.

4 Discussion

This paper presented a fuzzy generalization to the view-based semantic search paradigm. A prototype implementation and its application to a data set in semantic eHealth portal was discussed and evaluated.

4.1 Contributions

The presented fuzzy view-based search method provides the following benefits when in comparison with the crisp view-based search:

Ranking of the result set. Traditional view-based semantic search provides sophisticated means to order results by grouping. However, it does not provide ways to rank results. By extending the set theoretical model of view-based search to fuzzy sets, ranking the results is straightforward based on the membership functions of the concepts.

Enabling the separation of end-user views from annotation ontologies. In many cases the formal ontologies created by and for domain experts are not ideal for the end-user to search. The concepts are not familiar to a non-expert and the organization of the ontology may be unintuitive. In this paper we tackled the problem by creating a way to represent search views separately from the ontologies and to map the search concepts to the annotation concepts. The mappings may contain uncertainty.

No commitment to any particular implementation or weighting scheme. The paper presents a generic framework to include uncertainty and vagueness in view-based search. It can be implemented in many different ways, as long as the weighting or ranking methods can be mapped to fuzzy values in the range (0,1].

4.2 Related Work

The work in this paper generalizes the traditional view-based search paradigm [23, 11, 13] and its semantic extension developed in [13,21,12,17].

The problem of representing vagueness and uncertainty in ontologies has been tackled before. In methods using rough sets [28,22] only a rough, egg-yolk representation of the concepts can be created. Fuzzy logic [30], allows for a more realistic representation of the world.

Also probabilistic methods have been developed for managing uncertainty in ontologies. Ding and Peng [7] present principles and methods to convert an OWL ontology into a Bayesian network. Their methods are based on probabilistic extensions to description logics [15,8]. Also other approaches for combining Bayesian networks and ontologies exist. Gu [10] present a Bayesian approach for dealing with uncertain contexts. In this approach, probabilistic information is represented using OWL. Probabilities and conditional probabilities are represented using classes constructed for these purposes. Mitra [20] presents a probabilistic ontology mapping tool. In this approach the nodes of the Bayesian network represent matches between pairs of classes in the two ontologies to be mapped. The arrows of the BN are dependencies between matches.

Kauppinen and Hyvönen [14] present a method for modeling partial overlap between versions of a concept that changes over long periods of time.

Our method is based on fuzzy logic [30]. We have applied the idea presented by Straccia [27] in his fuzzy extension to the description logic *SHOIN(D)* and Bordogna [5] of using fuzzy implication to model fuzzy inclusion between fuzzy sets. Also other fuzzy extensions to description logic exist, such as [26,19].

Zhang et al. [31] have applied fuzzy description logic and information retrieval mechanisms to enhance query answering in semantic portals. Their framework is similar to ours in that both the textual content of the documents and the semantic metadata is used to improve information retrieval. However, the main difference in the approaches is that their work does not help the user in query construction whereas the work presented in this paper does by providing an end-user specific view to the search items.

Akrivas et al. [4] present an interesting method for context sensitive semantic query expansion. In this method, user's query words are expanded using fuzzy concept hierarchies. An inclusion relation defines the hierarchy. The inclusion relation is defined as the composition of subclass and part-of relations. Each word in a query is expanded by all the concepts that are included in it according to the fuzzy hierarchy.

In [4], the inclusion relation is of the form $P(a, b) \in [0, 1]$ with the following meaning: A concept a is completely a part of b. High values of the $P(a, b)$ function mean that the meaning of a approaches the meaning of b. In our work the fuzzy inclusion was interpreted as fuzzy implication, meaning that the inclusion relation itself is partial.

Widyantoro and Yen [29] have created a domain-specific search engine called PASS. The system includes an interactive query refinement mechanism to help to find the most appropriate query terms. The system uses a fuzzy ontology of term associations as one of the sources of its knowledge to suggest alternative query terms. The ontology is organized according to narrower-term relations. The ontology is automatically built using information obtained from the system's document collections. The fuzzy ontology of Widyantoro and Yen is based on a set of documents, and works on that document set. The automatic creation of ontologies is an interesting issue by itself, but it is not considered in our paper. At the moment, better and richer ontologies can be built by domain specialists than by automated methods.

4.3 Lessons Learned and Future Work

The fuzzy generalization of the (semantic) view-based search paradigm proved to be rather straight forward to design and implement. Crisp view-based search is a special case of the fuzzy framework such that the annotations and the mappings have the weight 1.0, i.e. are crisp.

Our preliminary evaluation of ranking search results with the framework were promising. However, the number of test subjects and the size of the test data set was still too small for proper statistical analysis.

Our framework did get some inspiration from fuzzy versions of description logics. We share the idea of generalizing the set theoretic basis of an IR-system to fuzzy sets in order to enable the handling of vagueness and uncertainty. In addition, the use of fuzzy implication to reason about fuzzy inclusion between concepts is introduced in the fuzzy version [27] of the description logic *SHOIN(D)*. However, the ontologies that we use are mainly simple concept taxonomies, and in many practical cases we saw it as an unnecessary overhead to anchor our framework in description logics.

Furthermore, the datasets in our *Tervesuomi.fi* eHealth portal case study are large. The number of search-items will be probably between 50,000 and 100,000, and the number of annotation concepts probably between 40,000 and 50,000. For this reason we wanted to build our framework on the view-based search paradigm that has proven to be scalable to relatively large data sets. For example, the semantic view-based search engine *OntoViews* was tested to scale up to 2.3 million search items and 275,000 search categories in [17]. The fuzzy generalization adds only a constant coefficient to the computational complexity of the paradigm.

In the future we intend to implement the framework with a larger dataset in the semantic *Tervesuomi.fi* eHealth portal and test it with a larger user group. The fuzzy framework will be attached to the *OntoViews* tool as a separate ranking module. Thus, there is not a need for major refactoring of the search engine in *OntoViews*. In addition we intend to apply the framework to the ranking of the recommendation links created by *OntoDella*, which is the semantic recommendation service module of *OntoViews*.

Acknowledgments. Our research was funded mainly by the National Technology Agency Tekes. The National Public Health Institute of Finland (NPHI) provided us with the data annotated by Johanna Eerola.

References

1. *RDF Primer*. http://www.w3.org/TR/rdf-primer.
2. *SKOS Mapping Vocabulary Specification*, 2004. http://www.w3.org/2004/02/skos/mapping/spec/.
3. *SKOS Core Guide*, 2005. http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/.
4. G. Akrivas, M. Wallace, G. Andreou, G. Stamou, and S. Kollias. Context - sensitive semantic query expansion. In *Proceedings of the IEEE International Conferrence on Artificial Intelligence Systems (ICAIS)*, 2002.
5. G. Bordogna, P. Bosc, and G. Pasi. Fuzzy inclusion in database and information retrieval query interpretation. In *ACM Computing Week - SAC'96*, Philadelphia, USA, 1996.
6. S. Decker, M. Erdmann, D. Fensel, and R. Studer. Ontobroker: Ontology based access to distributed and semi-stuctured information. *DS-8*, pages 351–369, 1999. http://citeseer.nj.nec.com/article/decker98ontobroker.html.
7. Z. Ding and Y. Peng. A probabilistic extension to ontology language owl. In *Proceedings of the Hawai'i Internationa Conference on System Sciences*, 2004.
8. R. Giugno and T. Lukasiewicz. P-shoq(d): A probabilistic extension of shoq(d) for probabilistic ontologies in the semantic web. INFSYS Research Report 1843-02-06, Technische Universität Wien, 2002.
9. C. Goble, S. Bechhofer, L. Carr, D. De Roure, and W. Hall. Conceptual open hypermedia = the semantic web. In *Proceedings of the WWW2001, Semantic Web Workshop*, Hongkong, 2001.
10. T. Gu and D.Q. Zhang H.K. Pung. A bayesian approach for dealing with uncertain contexts. In *Advances in Pervasive Computing*, 2004.
11. M. Hearst, A. Elliott, J. English, R. Sinha, K. Swearingen, and K.-P. Lee. Finding the flow in web site search. *CACM*, 45(9):42–49, 2002.
12. Eero Hyvönen, Eetu Mäkelä, Mirva Salminen, Arttu Valo, Kim Viljanen, Samppa Saarela, Miikka Junnila, and Suvi Kettula. Museumfinland – finnish museums on the semantic web. *Journal of Web Semantics*, 3(2):25, 2005.
13. Eero Hyvönen, Samppa Saarela, and Kim Viljanen. Application of ontology techniques to view-based semantic search and browsing. In *The Semantic Web: Research and Applications. Proceedings of the First European Semantic Web Symposium (ESWS 2004)*, 2004.
14. T. Kauppinen and E. Hyvönen. Geo-spatial reasoning over ontology changes in time. In *Proceedings of IJCAI-2005 Workshop on Spatial and Temporal Reasoning*, 2005.
15. D. Koller, A. Levy, and A. Pfeffer. P-classic: A tractable probabilistic description logic. In *Proceedings of AAAI-97*, 1997.
16. A. Maedche, S. Staab, N. Stojanovic, R. Struder, and Y. Sure. Semantic portal - the seal approach. Technical report, Institute AIFB, University of Karlsruhe, Germany, 2001.
17. Eetu Makelä, Eero Hyvönen, Samppa Saarela, and Kim Viljanen. Ontoviews – a tool for creating semantic web portals. In *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004)*, May 2004.

18. A. Maple. Faceted access: a review of the literature, 1995. http://library.music.indiana.edu/tech_s/mla/facacc.rev.
19. M. Mazzieri and A. F. Dragoni. Fuzzy semantics for semantic web languages. In *Proceedings of ISWC-2005 Workshop Uncertainty Reasoning for the Semantic Web*, Nov 2005.
20. P. Mitra, N. Noy, and A.R. Jaiswal. Omen: A probabilistic ontology mapping tool. In *Working Notes of the ISCW-04 Workshop on Meaning Coordination and Negotiation*, 2004.
21. Eetu Mäkelä, Eero Hyvönen, and Teemu Sidoroff. View-based user interfaces for information retrieval on the semantic web. In *Proceedings of the ISWC-2005 Workshop End User Semantic Web Interaction*, Nov 2005.
22. J. Pawlak. Rough sets. *International Journal of Information and Computers*, 1982.
23. A. S. Pollitt. The key role of classification and indexing in view-based searching. Technical report, University of Huddersfield, UK, 1998. http://www.ifla.org/IV/ifla63/63polst.pdf.
24. A. Rector. Defaults, context, and knowledge: Alternatives for owl-indexed knowledge bases. In *Proceedings of Pacific Symposium on Biocomputing*, 2004.
25. G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA, 1987.
26. G. Stoilos, G. Stamou, V. Tzouvaras, J. Pan, and I. Horrocks. The fuzzy description logic f-shin. In *Proceedings of ISWC-2005 Workshop Uncertainty Reasoning for the Semantic Web*, Nov 2005.
27. Umberto Straccia. Towards a fuzzy description logic for the semantic web (preliminary report). In *2nd European Semantic Web Conference (ESWC-05)*, number 3532 in Lecture Notes in Computer Science, pages 167–181, Crete, 2005. Springer Verlag.
28. H. Stuckenschmidt and U. Visser. Semantic translation based on approximate re-classification. In *Proceedings of the 'Semantic Approximation, Granularity and Vagueness' Workshop*, 2000.
29. D.H. Widyantoro and J. Yen. A fuzzy ontology-based abstract seawrch engine and its user studies. In *The Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, 2002.
30. L. Zadeh. Fuzzy sets. *Information and Control*, 1965.
31. L. Zhang, Y. Yu, J. Zhou, C. Lin, and Y. Yang. An enhanced model for searching in semantic portals. In *Proceedings of the Fourteenth International World Wide Web Conference*, May 2005.
32. H.-J. Zimmermann. *Fuzzy Set Theory and its Applications*. Springer, 2001.

A Semantic Search Conceptual Model and Application in Security Access Control[*]

Kunmei Wen, Zhengding Lu, Ruixuan Li, Xiaolin Sun, and Zhigang Wang

Internet and Distributed Computing Lab,
College of Computer Science and Technology,
Huazhong University of Science and Technology,
Wuhan 430074, Hubei, P.R. China
kunmei.wen@gmail.com

Abstract. We propose a conceptual model for semantic search and implement it in security access control. The model provides security access control to extend the search capabilities. The scalable model can integrate other ontology providing the general ontology as the transformation interface. We combine text Information Retrieval (IR) with semantic inference in the model. So it can not only search the resources and the relationships between them according to the user's privileges, but also locate the exact resource using text IR. We build a security ontology based on Role-Based Access Control (RBAC) policy. A semantic search system Onto-SSSE is implemented based on the model. The system can perform some complex queries using ontology reasoning, especially about association queries such as the relationships between resources. The evaluation shows that the new system performs better than exiting methods.

1 Introduction

Semantic Web [1] proposed by Tim Berners-Lee is the next generation of web portals. The aim is to annotate all the resources on the web and establish all kinds of semantic relationships between them understandable for the machine. As the most important application of Semantic Web, semantic search is being got more and more attention. The concept of semantic search is put forward in [2]. Semantic search integrates the technologies of Semantic Web and search engine to improve the search results gained by current search engines and evolves to next generation of search engines built on Semantic Web.

Semantic search finds out the semantic information by means of inferring internal knowledge in Knowledge Base (KB). Description Logic (DL) [3,4] is well known as the base of ontology language such as Web Ontology Language (OWL) [5]. All modern DL system are implemented based on tableaux algorithm [6], many optimized technologies [7] are explored. [8] defines the search object

[*] This work is partially supported by National Natural Science Foundation of China under Grant 60403027, Natural Science Foundation of Hubei Province under Grant 2005ABA258, Open Foundation of State Key Laboratory of Software Engineering under Grant SKLSE05-07, Huawei Foundation.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 366–376, 2006.
© Springer-Verlag Berlin Heidelberg 2006

of semantic search. One is searching the Internet. The other is searching the Semantic Web portals. Semantic Web portals are composed of domain ontology and KB. An enhanced model for searching in semantic portals is proposed in [9]. The model combines the formal DL and fuzzy DL [10] to implement the integration of information retrieval and structure query.

Ranking the search results [11,12] is the key technology of semantic search. Since it is expected that the number of relationships between entities in a KB will be much larger than the number of entities themselves, the likelihood that Semantic Association searches would result in an overwhelming number of results for users is increased, therefore elevating the need for appropriate ranking schemes. In [13], a method is proposed to rank the results according to the important values of web resources based on the technology of modern IR [14]. The ranking method in [15] focuses on the semantic metadata to find out the complex relationships and predict the user's requirement to distinguish semantic associations.

Role-based access-control (RBAC) models show clear advantages over traditional discretionary and mandatory access control models with regard to these requirements. There has been a lot of research into the various ways of specifying policy in the last decade. One of them is ontology-based approach. Some initial efforts in the use of Semantic Web representations for basic security applications such as access control of policy have begun to bear fruit. KAoS [16] and Rei [17] are semantic policy languages represented in OWL to specify security policy supporting positive and negative authorization and obligation. The reasoning of KaoS policy is based on DL, which limits the expressive power of policy, as DL doesn't support rule now. As to Rei, it doesn't support the model of RBAC96 explicitly. Besides, they can't intuitively specify the important security principle, separation of duty.

There are great demands for this kind of semantic search considering security issues, such as Intranet search which must satisfy access control request in the background of government or business. We propose a semantic search model that enables the user to find his resources based on his privilege. The proposed model combines text IR with semantic inference. Based on the model a semantic search system Onto-SSSE is implemented and evaluated.

The rest of the paper is organized as follows. We present the architecture of the semantic search model and discuss the components of the model and the relationships between components in section 2. The third section discusses the integration of search and inference to get the semantic information and presents the ranking method in semantic search. After that in the forth section the security ontology based on RBAC [18] policy is introduced and instances are described. In section 5 experiment and evaluation are carried out. Related work is introduced in section 6. Section 7 contains conclusions and future work.

2 Architecture of Semantic Search Model

In this section we propose the architecture of the semantic search conceptual model. The architecture of the model is shown in Fig.1. The components of the model and the relationships between them are described as follows.

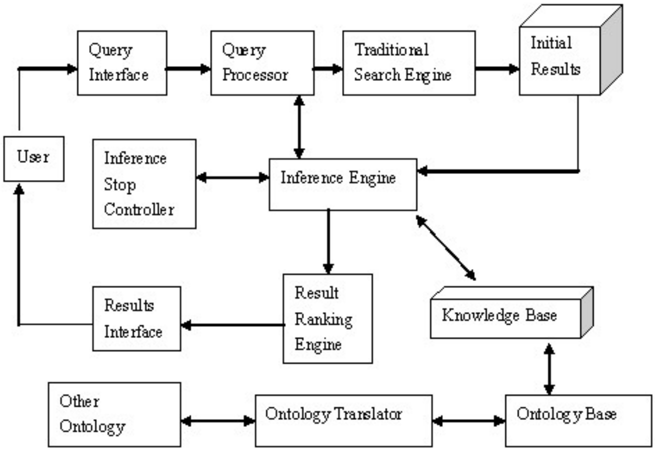


Fig. 1. Architecture of the proposed semantic search conceptual model

Query Interface receives the queries from users. The query is defined as keywords or formal queries. *Query Processor* converts user's queries to uniform format which is defined by the model. Then these queries will be distributed in two ways. One is forwarded to a traditional search engine. The other is forwarded to an inference engine. By means of the operation of *Traditional Search Engine*, we will get the Initial Results using text IR technology. The initial search results are also transformed to inference engine. If the user submits a formal query, then the query will push directly to inference engine. *Knowledge Base* restores domain ontology and reasoning rules or knowledge and is the base for reasoning. *Inference engine* performances the operation of reasoning to get the semantic information and obtains all the search results. *Inference Stop Controller* decides how much to reason and when the reasoning should stop. *Result Ranking Engine* ranks all the results returned by the inference engine. Finally user gets the results through *Results Interface.*

The rest three modules are *Other Ontology*, *Ontology Translator* and *Ontology Base*. They are used to expend the capabilities of semantic search and implement the scalability of the model.

3 Semantic Search Model

The semantic search model mainly is made up of three parts: definition of query form, reasoning based on description logic and result ranking.

3.1 Definition of Query Form

Different users have different privileges for different resources. Some users have the privileges to see or edit or delete the resources such as web pages or news, while others have not the privileges to browse them. Only after assuring that the user has the right privilege, we could return the resources back to the user through traditional IR technology.

A query is defined as the form $Q_i = Q_{i1} \cap Q_{i2} \cap Q_{i3}$ the semantic search model. Here Q_{i1} means user or role, Q_{i2} is any formal query about resources or the relation-ships between them and Q_{i3} is a keyword query. If Q_{i1} is not appear, that means the user has the default privilege. Q_{i1} and Q_{i2} are implemented based ontology reasoning while Q_{i3} is carried out through traditional text IR technology.

So there are five typical queries as follows:

Q_{i11}: User Query, form as Q_{i11} = "A" where A means a user.

Q_{i12}: Role Query, form as Q_{i12} = "B" where B means a role. In fact, Q_{i11} and Q_{i12} belong to concept query Q_{i1}, so we can get $Q_{i1} = Q_{i11}$ or Q_{i12} = "C" where C means a concept.

Q_{i2}: Relationship Query, form as $Q_{i2} = "C1"\&"C2"$ where C1 and C2 are concepts.

Q_{i3}: Keyword Query, form as Q_{i3} = "D" where D means a keyword which appears in the text. In fact, Q_{i3} belongs to traditional query.

$Q_{i1} \cap Q_{i3}$: Conjunctive Query, form as $Q_{i1} \cap Q_{i3} = ("A" or "B")"D"$ where A means a user, B means a role and D means a keyword.

3.2 Reasoning Based on Description Logic

We implement four kinds of reasoning based on Description Logic in the semantic search model. The architecture of the Knowledge Base based on Description Logic is showed in Fig.2.

The first is *Role Activation Reasoning.* Given $Q_i = Q_{i11}$ where Q_{i11} means user, we can get all the roles the user has. For example if Alice is a user and she can act as Direct or ProjectLeader, then we get all her roles through role activation reasoning.

The second is *Role Privilege Reasoning.* Given $Q_i = Q_{i12}$ where Q_{i12} means role, we can get all the sub-roles of the role and then get all the privileges from these roles. For example if we get role ProjectLeader, through role privilege reasoning we can get the sub-roles including ProductionEngineer and QualityEngineer, so Project-Leader should have all the privileges both ProductionEngineer and QualityEngineer have.

The third is *Relationship Reasoning.* Given $Q_i = Q_{i2}$ where Q_{i2} includes two concepts, we can get the relationship between them or null if there is not any relation-ship. For example if ProjectLeader is the senior role of ProductionEngineer, given the query ProjectLeader & ProductionEngineer, we should be returned the result seniorRoleOf.

The forth is *Conjunctive Query Reasoning.* In fact it integrates inference with search by providing both formal query and keyword query. Given $Q_i = Q_{i1} \cap Q_{i3}$ where Q_{i1} means user or role and Q_{i3} is a keyword query, the semantic search model firstly performance Q_{i1} to judge the user's or the role's privilege. If the user or the role has the corresponding privilege the model carries out Q_{i3} to locate the exact resource. So it can not only locate the exact place of the resource

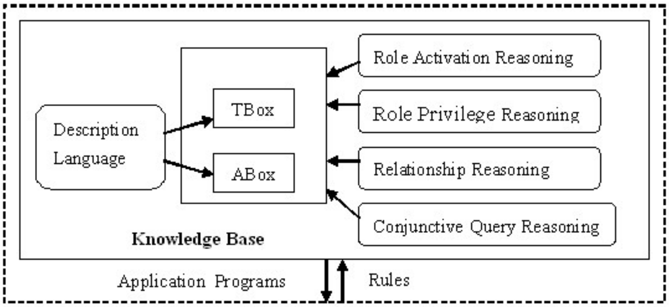


Fig. 2. Architecture of the Knowledge Base based on Description Logic

using the traditional text IR but also implement security access control through inference.

3.3 Result Ranking

Ranking the search results is very important for the implementation of semantic search. It is possible that the number of relationships between entities in a KB will be much larger than the number of entities themselves. We provide a ranking scheme based on the ranking value. The Ranking value for the query Q_i is defined as the form $R_i = R_{i1} + R_{i2} + R_{i3}$ for the query $Q_i = Q_{i1} \cap Q_{i2} \cap Q_{i3}$. Here R_{i1} is the ranking value for Q_{i1}, at the same time Ri2 is the value for Q_{i2} and R_{i3} is that for Q_{i3}. The reasoning result is used to compute the values of R_{i1} and R_{i2}. Given Q_{i1}, if the user has the privilege for the resource, then the value of R_{i1} is 1. Otherwise it is 0. If $R_{i1} = 0$ then $R_{i2} = R_{i3} = 0$. That means if the user has no corresponding privilege he will not be permitted to do any operation on the resource, in this case $R_i = 0$.

For R_{i2}, it is possible to return many relationships between two concepts. So the value R_{i2} is determined by the important value of the relationship. For every relation-ship in domain ontology we define an important value Ii which is between o and 1. So it is reasonable to get the conclusion $R_{i2} = I_i$.

R_{i3} is corresponding to Q_{i3}. Searching is used to locate the resource through key-word query. Therefore we can use traditional tf-idf method to compute the value of R_{i3}.

4 RBAC Security Ontology and Description of Instances

KAoS and Rei mentioned above are semantic policy languages represented in OWL to specify security policy. The reasoning of KaoS policy is based on DL. As to Rei, it only supports the rule. KAoS and Rei don't support the recursive authorization. Be-sides, they can't intuitively specify the important security principle, separation of duty (SoD). RBAC is a popular security policy. Here we assume that the readers are familiar with the RBAC policy. We build a security ontology shown in Fig.3 based on RBAC policy.

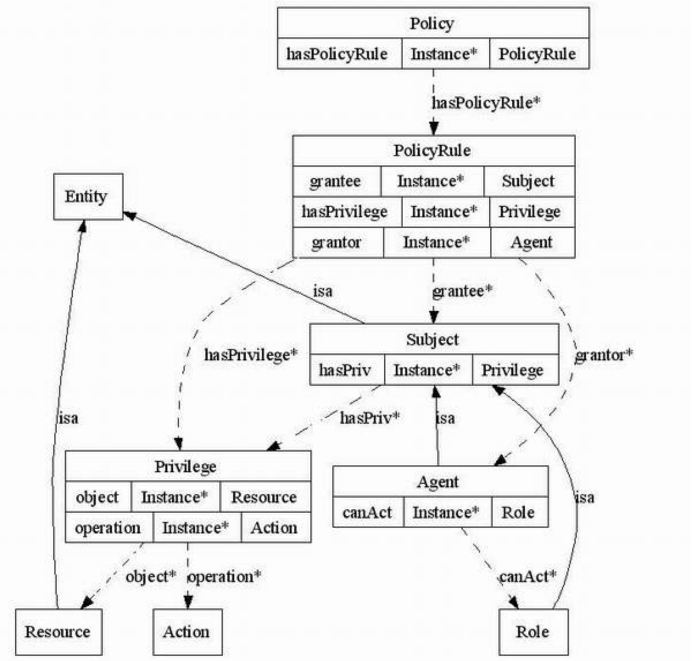


Fig. 3. RBAC security ontology

In RBAC security ontology, nine basic classes are created. They are *Policy*, *PolicyRule*, *Priviledge*, *Entity*, *Resource*, *Agent*, *Subject*, *Role* and *Action*. We give properties for these classes, for example on the top of the figure 3 *hasPolicyRule* is the property of the class *policy*. The right side of the property is its range, for example the range of the property *grantor* is the instances of the class *Agent* and its domain is the class *PolicyRule*. The arrow between two classes indicates the relationships between them. Real line is the subsumption relationship while dashed one defines the property between them. For example *subject* is a subclass of entities, so the relationship between them is "isa". From the figure 3, we can see there are relationships between these classes: *PolicyRule*'s grantee is *Subject*, its grantor is *Agent* and it has *Privilege*; both of *Agent* and *Role* are subclasses of *Subjects*, at the same time *Subject* and *Resource* are subclasses of Entities; *Privilege*'s object is *Resource* and its operation is *Action*. *Agent* can act as *role* where we can think *Agent* has the same meaning with user.

We use OWL DL as our ontology language. As one of W3C's standards, OWL DL is widely used in application. Here is the example fragment of the owl language building the security ontology, showing as the follows:

$< owl : Classrdf : ID = "Subject" >< rdfs : subClassOf >$
$< owl : Classrdf : ID = "Entity" / >< /rdfs : subClassOf >$
$< /owl : Class >< owl : Classrdf : ID = "Role" >$
$< rdfs : subClassOfrdf : resource = "\#Subject" / > ..$

$< /owl : ObjectProperty >< owl : ObjectPropertyrdf : ID = "hasPriv" >$
$< rdfs : rangerdf : resource = "\#Privilege" / >$
$< rdfs : domainrdf : resource = "\#Subject" / >< /owl : ObjectProperty >$

The reason that we choose RBAC policy as our ontology is that the RBAC is more general than other security policies. It is easy to transform other policies to RBAC policy, such as *Mandatory Access Control* and *Discretionary Access Control*, while the reverse transform is not possible. It means that RBAC security ontology is appropriate to be a uniform security policy interface.

To illustrate semantic search more clearly, we give role instances hierarchy graph shown in figure4 and simple privilege instances graph showed in figure 5. From the Fig.4 we can see that Director is the most high-level role.

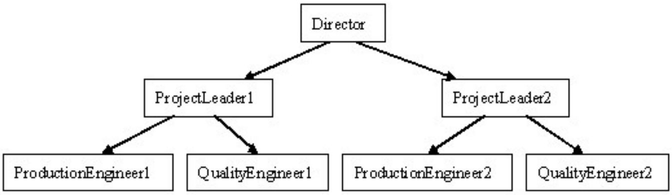


Fig. 4. RBAC security ontology

We create some instances for classes such as roles, agents and resources. There are six roles including *director, ProjectLeader1, ProductionEngineer1, QualityEngineer1, ProjectLeader2, ProductionEngineer2* and *QualityEngineer2.* There are two subclasses of resources resources1 and resources2. We define resource1 two instances webpage11 and webpage12, define resource2 two instances webpage21 and webpage22. We define only one instance "browse" for Action.

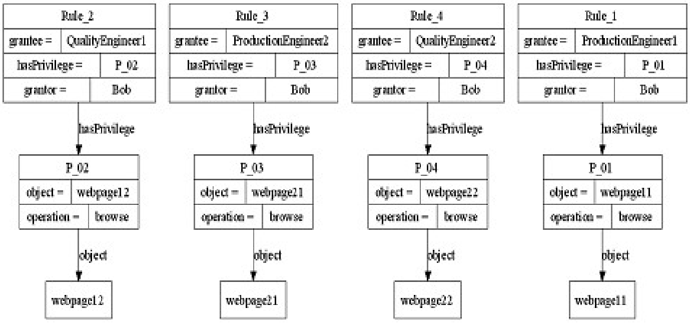


Fig. 5. Simple privilege instances graphy

There are application privileges shown in Fig.5. For example ProductionEngineer1 can browse the resource webpage11 which belong to resource1 and ProductionEngineer2 can browse the resource webpage21 which belong to resource2.

5 Experiment and Evaluation

We implement Ontology Security Semantic Search Engine (Onto-SSSE) in Java. We used the Lucene [19] search engine as the traditional search engine based on key-word query and Jena as the reasoning tool based on RBAC security ontology. We do some experiments on Onto-SSSE. The Table 1 shows the search results for some typical queries.

Table 1. Semantic search results

Query ID	Query form	Query form	Reasoning Type	Query Result
Q_1	Q_{i11}	“Alice”	Role Activation Reasoning	Director, ProjectLeader1
Q_2	Q_{i12}	“Director”	Role Privilege Reasoning	Sub-roles:ProjectLeader1,ProductionEngineer1, QualityEngineer1,ProjectLeader2, ProductionEngineer2,QualityEngineer2; Privileges:(browse,webpage11), (browse,webpage12), ……
Q_3	Q_{i2}	“ProjectLeader1”& “ProductionEngineer1”	Relationship Reasoning	seniorRoleOf
Q_4	Q_{i3}	“computer”	No Reasoning	Null (no privilege)
Q_5	$Q_{i1}\cap Q_{i3}$	“Director & computer”	Conjunctive Query Reasoning	webpage list: webpage11, webpage21…. Where include the text “computer” in these web pages

Q_1 is a simple query just for the user. $Q_1 =$ ”Alice”. The result is Director and ProjectLeader1, because they are the roles as which Alice can act. Q_2 is a query for the role $Q_2 =$ ”Director”, we are returned all the sub-roles of Director and all the privilege these roles have. Director has six sub-roles such as ProjectLeader1 and ProductionEngineer1; Director has the privilege (browse, webpage11), (browse, webpage12) and so on. Q_3 is a query for the relationship. The results is seniorRoleOf between ProjectLeader1 and ProductionEngineer1.

Q_4 is the simple keyword query, because the default user or role has no required privilege, so Null is returned. Q_5 is a Conjunctive Query as the form "Director & computer", the result returned is the webpage list where the pages include the text "computer".

As pointed out in [20], currently there is no commonly agreed evaluation method-ology and benchmark for semantic search. We constitute our research group's evaluation dataset. The results are analyzed positively in 90%. The dataset is made up of the RBAC security ontology (including 12 classes, 16 properties and 20 individuals) and the set of campus web pages (more then 200MB). We mainly compare our system with traditional method based on keyword query shown in Table 2. From the table, we can find that the new semantic search system performs better than traditional one especially about the reasoning function.

Table 2. Compare between the semantic search with traditional method

Query form	Reasoning Type	Traditional method	Semantic search
Q_{i11}	Role Activation Reasoning	Not support	Support
Q_{i12}	Role Privilege Reasoning	Not support	Support
Q_{i2}	Relationship Reasoning	Not support	Support
Q_{i3}	No Reasoning	Support	Support
$Q_{i1} \cap Q_{i3}$	Conjunctive Query Reasoning	Not support	Support

6 Related Work

Tap Knowledge Base (KB) [21] is implemented by Stanford University, IBM and other research institutions. Tap KB brings Semantic Web technology into Google to improve the search efficiency through providing additional results. The two kinds of different results are shown on the same page. However the search object is still the traditional resource, not the one on Semantic Web. The method only responds the keyword query, not supporting the form query, so it could not integrate information retrieval and formal semantic query tightly. [22] provides an ontology-based information retrieval model to support result ranking. The method transforms the key-word query to structure query, not combining them.

Swoogle, a prototype system of IR is provided in [23]. The search results are physical documents on Semantic Web (such as RDF and OWL files). However Swoogle has not used the semantic structure information in documents. When the large documents are queried, the useful information is very little and user need analyze the whole file to locate the semantic information.

Turing center in the University of Washington develops the system KnowItAll [24] to extract the information on the Web. [25] prefer some methods of information extraction to search the Web and build up domain KB. Its long-term aim is to re-place the search engine by information extraction. This is another kind of semantic search.

7 Conclusions and Future Work

In this paper we propose a conception model for semantic search and apply it in security access control domain. We combine text IR with semantic reference in the model. The model extends the search capabilities of existing methods through implementing security access control. It also can answer some complex queries such as the relationships between resources. A semantic search system is implemented based on the model. The evaluation shows that the new system performs better than the exiting methods.

We plan to get improvement in the following three aspects. The first is to perform search in a larger dataset. The second is to improve the reasoning efficiency. The reasoning efficiency can not satisfy the user.

References

1. T.Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. Scientific American, May 2001
2. Guha R, McCool R, Miller E. Semantic search. Proceeding of the 12th International World Wide Web Conference. Budapest, Hungary, May 2003: 700-709
3. Franz Baader, Deborah McGuinness, Daniele Nardi, et al. The Description Logic Hand-book: Theory, Implementation and Applications, Cambridge, UK: Cambridge Univ. Press, 2003
4. D. Calvanese, G. Giacomo, and M. Lenzerini. Ontology of Integration and Integration of Ontologies. In Description Logic Workshop 2001: 10-19
5. Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From SHIQ and RDF to OWL: The Making of A Web Ontology Language. J. of Web Semantics, 2003, 1(1):7-26
6. Ian Horrocks and Ulrike Sattler. A Tableaux Decision Procedure for SHOIQ. In Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI), 2005
7. F. Baader and U. Sattler. An Overview of Tableau Algorithms for Description Logics. Studia Logica, 2001, 69:5-40
8. A. Sheth, C. Bertram, D. Avant, B. Hammond, K. Kochut, and Y. Warke. Managing Se-mantic Content for the Web. IEEE Internet Computing, 2002, 6(4)
9. Lei Zhang, Yong Yu, Jian Zhou, Chenxi Lin, Yin Yang: An Enhanced Model for Searching in Semantic Portals. WWW 2005: 453-462
10. U. Straccia. Reasoning Within Fuzzy Description Logics. Journal of Artificial Intelligence Research, 2001(14)
11. N. Stojanovic, R. Studer, and L. Stojanovic. An Approach for the Ranking of Query Re-sults in the Semantic Web. In Proc. of ISWC 2003
12. Anyanwu, K., Maduko, A., and Sheth, A.P.: SemRank: Ranking Complex Relationship Search Results on the Semantic Web, Proceedings of the 14th International World Wide Web Conference, ACM Press, 2005
13. Bhuvan Bamba, Sougata Mukherjea: Utilizing Resource Importance for Ranking Semantic Web Query Results. SWDB 2004: 185-198
14. Baeza-Yates and Ribeiro-Neto. Modern Information Retrieval. Addison Wesley 1999
15. Boanerges Aleman-Meza, Christian Halaschek-Wiener, I. Budak Arpinar, Cartic Rama-krishnan, Amit P. Sheth, Ranking Complex Relationships on the Semantic Web, IEEE Internet Computing, 2005, 9(3): 37-44

16. A. Uszok, J. Bradshaw, R. Jeffers, et al. KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction, and Enforcement. IEEE 4th International Workshop on Policies for Distributed Systems and Networks, 2003
17. L. Kagal, T. Finin, and A. Joshi. A Policy Language for Pervasive Systems. Fourth IEEE International Workshop on Policies for Distributed Systems and Networks, 2003
18. Ravi S. Sandhu, Edward J. Coyne et al. Role-Based Access Control models. IEEE Com-puter, 1996, 29(2): 38-47
19. Lucene Search Engine. http://jakarta.apache.org/lucene
20. C. Rocha, D. Schwabe, and M. P. de Arag ao. A Hybrid Approach for Searching in the Semantic Web. In Proc of WWW 2004: 374-383
21. Guha, R., McCool, R.: TAP: A Semantic Web Test-bed. Journal of Web Semantics, 2003, 1(1)
22. Vallet D, Fernmndez M , Castells P. An Ontology-based Information Retrieval Model. 2nd European Semantic Web Conference (ESWC). Heraklion, Greece, May 2005
23. Ding L , Finin T, Joshi A, et al. Swoogle: A Search and Metadata Engine for the Semantic Web. In CIKM'04. Washington DC, USA, November 2004
24. Michael Cafarella, Doug Downey, Stephen Soderland, and Oren Etzioni. KnowItAll: Fast, Scalable Information Extraction from the Web. Proceedings of the Conference on Empiri-cal Methods in Natural Language Processing EMNLP 2005
25. Ana-Maria Popescu and Oren Etzioni, Extracting Product Features and Opinions from Reviews, Proceedings of the Conference on Empirical Methods in Natural Language Proc-essing EMNLP 2005

Document Filtering for Domain Ontology Based on Concept Preferences

Bo-Yeong Kang and Hong-Gee Kim

Biomedical Knowledge Engineering Laboratory
Dentistry College, Seoul National University
Yeongeon-dong, Jongro-gu, Seoul, Korea

Abstract. For domain ontology construction and expansion, data-driven approaches based on web resources have been actively investigated. Despite the importance of document filtering for domain ontology management, however, few studies have sought to develop a method for automatically filtering out domain-relevant documents from the web. To address this situation, here we propose a document filtering scheme that identifies documents relevant to a domain ontology based on concept preferences. Testing of the proposed filtering scheme with a business domain ontology on 1,409 YahooPicks web pages yielded promising filtering results that outperformed the baseline system.

1 Introduction

Ontology has a long history in philosophy, where it refers to the notion of existence. We can think of a lexicon as a simple type of ontology. From a computational viewpoint, a lexicon can generally be looked upon as an ontology if it is organized with a machine-readable specification for concepts and relationships. Recent research into ontology management has spanned various fields, including information retrieval, biomedical informatics, e-commerce and intelligent internet technologies. The main approach used to date has been a top-down method whereby domain experts construct or manage the ontologies manually [1]. Although data-driven methods that learn and manage a domain ontology by analyzing a large quantity of resources have been examined, most previous studies have used domain experts in constructing the ontology or have exploited manually constructed domain resources [2]. Such approaches have the shortcomings that they require manual construction of the domain ontology and domain resources, which is labor-intensive and time-consuming. To address this issue, in the present work we sought to develop a method for identifying domain-relevant documents within large sets of web resources, which can serve as the basis for domain ontology development and management.

Document filtering has been actively investigated, with most studies focusing on collaborative filtering and content-based filtering approaches. Collaborative filtering uses feedback from multiple users to recommend the related service to users with similar interests [4,5,6]. Content-based filtering systems exploit the method that analyzes the relationship between the contents of a document and

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 377–386, 2006.
© Springer-Verlag Berlin Heidelberg 2006

the user's response to documents for identifying the user's interest on a topic [7,8]. Although various approaches have been developed for document filtering, to our knowledge few studies have examined the problem of filtering documents relevant to a domain ontology. In addition, traditional filtering methods have limitations that hinder their direct application to text filtering for ontology management; hence a new method that can handle domain ontologies instead of user profiles is required. Therefore, here we propose a concept preference-based filtering scheme that filters the documents that are relevant to a domain ontology.

This paper is organized as follows. In Section 2, we present our document filtering methodology based on concept preference, and in Section 3 we present the results of, and compare, experiments using the proposed methodology and a baseline system. Our conclusions are given in Section 4.

2 Document Filtering for Domain Ontology Based on Concept Preference

2.1 Intuition and Overview

Documents generally contain various terms, and in many cases the meaning of one term in a document can be comprehended by examining the other terms used in the same context as the term under consideration [9]. In accordance with the accepted view in computational linguistics that terms co-occurring with a particular term can provide a good representation of the meaning of that term [9], we look on the terms co-occurring with a particular concept in a document as features to represent the meaning of that concept. In practice, we define the association value between a concept and a co-occurring term as the concept preference, which represents the degree to which the co-occurring term expresses the meaning of the concept. Here, a concept denotes a class within the domain ontology.

Figure 1 shows a schematic overview of the proposed filtering methodology. When applied to a seed domain ontology constructed by domain experts, the proposed method first retrieves the top N web pages relevant to a given concept within the seed ontology using Google, and then calculates the concept preferences using terms in the retrieved web pages. Then, the system estimates the relevance degree of a given document to the domain ontology by reference to the calculated concept preference value, and finally recommends the documents relevant to the domain to the domain experts.

2.2 Concept Preference Modeling

To construct the training set for the concept preference calculation, the web page set for each concept in the seed ontology is collected by means of a Google search. For each concept within the seed ontology, the system processes a Google search to retrieve the relevant web pages, and takes the top N pages as the training set for the preference calculation. For example, the concept Stock extracted from an ontology is applied to Google as the query "stock", and then the top 10 web

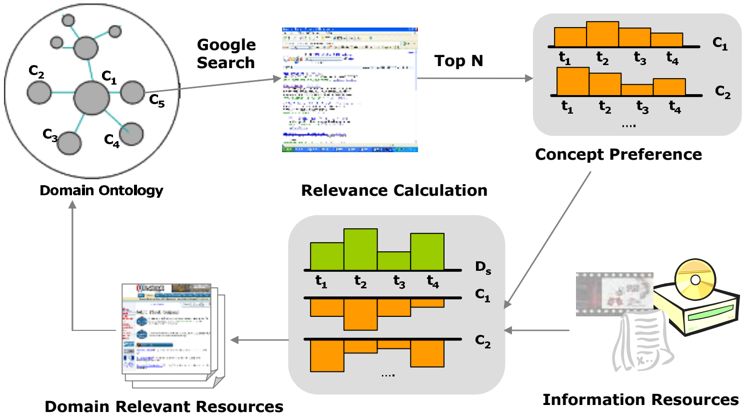


Fig. 1. Procedure of the document filtering relevant to domain ontology

pages are collected as the web page set for the concept preference calculation for the concept Stock.

Given the web page set constructed, the concept preference is calculated using the terms found in the web page set for each concept. As discussed in Section 2.1, we utilize the terms co-occurring with a particular concept in the web page set as features for representing the meaning of that concept. Then, the association value between a concept and a co-occurring term is defined as the concept preference, which represents the degree to which the co-occurring term expresses the meaning of the concept. Here we define the concept preference as follows.

Definition 1 (Concept Preference, CP). *The concept preference of a term for a given concept is defined as the degree to which the term expresses the meaning of the concept, and is calculated as an association measure between the term and the concept. The association between a term and a concept is represented by the degree to which the term co-occurs with the concept, as determined using the dice coefficient.*

$$CP(C_i, T_j) = \frac{2 \cdot f(C_i, T_j)}{f(C_i) + f(T_j)}, \ C_i \in O \text{ and } T_j \in T \quad (1)$$

where C_i corresponds to concept i in domain ontology O; T_j denotes term j in the set of terms T extracted from the web page set for concept C_i; $f(C_i, T_j)$ denotes the co-occurrence frequency of concept C_i and term T_j; and $f(C_i)$ and $f(T_j)$ denote the frequencies of concept C_i and term T_j in the web page set, respectively.

Example 1. In the top 10 web pages retrieved for the concept Enterprise, let the frequency of the concept Enterprise be 10, that of the term Intelligent be 10, and the co-occurrence frequency of Enterprise and Intelligent be 5. Then, the

CP of the term Intelligent for the concept Enterprise is determined to be 0.5 as follows:

$$\begin{aligned} CP(Enterprise, Intelligent) &= \frac{2 \cdot f(Enterprise, Intelligent)}{f(Enterprise) + f(Intelligent)} \\ &= \frac{2 \cdot 5}{10 + 10} = 0.5 \end{aligned}$$

Based on the notion of concept preferences in Definition 1, we define a concept preference set (Definition 2) that contains all of the co-occurring terms and their concept preferences for a particular concept. The concept preference set is defined as follows.

Definition 2 (Concept Preference Set, CPS). *The concept preference set for a given concept is defined as the set of pairs of co-occurring terms and their CP values.*

$$CPS(C_i) = \{(T_j, CP(C_i, T_j)) | CP(C_i, T_j) > 0,\ C_i \in O,\ T_j \in T\} \qquad (2)$$

Example 2. For a particular concept in the ontology, Stock, let us suppose it has the co-occurring terms Organization and Fallen, and that their CP values are 0.1 and 0.3, respectively. Then the CPS of the concept Stock is as follows:

CPS(Stock) = {(Organization, 0.1), (Fallen, 0.3)}

To represent a complex concept in the ontology, compound-noun concepts comprised of several terms are often used. Each single term within a compound-noun concept becomes an individual concept that represents one aspect of the meaning of the compound noun. In computational linguistic research, compound nouns have received a great deal of attention [10]; in particular, a framework for the lexical semantic representation has been developed in the Generative Lexicon (GL) theory [10]. Within GL model of lexical representation, the semantic content of a particular term is represented by four feature structures: type, argument, event and qualia structures [10]. Each structure contains several features and their values, which together express the semantic content of the term. To represent the semantic content of a compound noun, the features in the lexical representation of each of the terms that make up the compound noun are inherited and embedded within a lexical representation of the compound noun.

Based on this notion of compound noun concepts, we regard the meaning of a compound noun concept as being determined by the terms inherited from feature terms that are used to represent the meaning of each individual concept. Therefore, we define the $\oplus$ operation (Definition 3) to represent the composition process for the feature inheritance from the concept preference sets of individual concepts to the concept preference set of a compound-noun concept.

Definition 3 (Inheritance Operation $\oplus$). *The $\oplus$ operation between two CPSs generates a set of pairs composed of terms in each CPS and their maximum CP values, as follows:*

$$\begin{aligned}&CPS(C_m) \oplus CPS(C_n)\\ &= \{(T_p, max(CP(C_m, T_p), CP(C_n, T_p)))|\ C_m, C_n \in O,\\ &\qquad T_p \in CPS(C_m) \cup CPS(C_n)\}\end{aligned} \tag{3}$$

where C_m and C_n correspond to concept m and n in domain ontology O; T_p denotes term p, which constitutes of the set, $CPS(C_m) \cup CPS(C_n)$.

By the definition of the $\oplus$ operation, the concept preference set for a compound-noun concept is defined as follows.

Definition 4 (Compound-Noun Concept Preference Set). *The concept preference set for a compound-noun concept is derived from the concept preference sets of the individual concepts within the compound-noun concept using the inheritance operation, $\oplus$.*

$$\begin{aligned}CPS(C_i) &= CPS(C_{ik}) \oplus CPS(C_{il})\\ &= \{(T_q, max(CP(C_{ik}, T_q), CP(C_{il}, T_q)))|\ C_{ik}, C_{il} \in O,\\ &\qquad T_q \in CPS(C_{ik}) \cup CPS(C_{il})\}\end{aligned} \tag{4}$$

where C_{ik} and C_{il} correspond to individual concepts ik and il that constitute the compound-noun concept C_i in domain ontology O; T_q denotes term q that constitutes of the set, $CPS(C_{ik}) \cup CPS(C_{il})$.

Example 3. For the compound noun concept StockCompany, the CPS of StockCompany is derived as follows using the CPSs of the individual concepts Stock and Company, based on the $\oplus$ operation. The terms to represent the concept preferences of each individual concept, Stock and Company, are inherited as the feature terms used to express the concept preferences of the compound noun concept, StockCompany. Then the maximum CP value between a term and each of individual concepts is taken as the CP value of the term for the compound-noun concept.

CPS for an individual concept
  CPS(Stock) = {(Organization, 0.1), (Fallen, 0.3)}
  CPS(Company) = {(Organization, 0.4), (Fallen, 0.1), (S/W, 0.2)}
CPS of the compound noun concept, StockCompany
  CPS(StockCompany)
  = CPS(Stock) $\oplus$ CPS(Company)
  = {(Organization, 0.4), (Fallen, 0.3), (S/W,0.2)}

2.3 Concept Preference Based Document Filtering

This section illustrates the technique for identifying documents in information resources based on CP values determined as outlined in Section 2.2. As shown below, CP_{matrix} reflects the preference specification of the domain ontology for the documents.

$$CP_{matrix} = \begin{bmatrix} CP(C_1,T_1) & CP(C_1,T_2) & CP(C_1,T_3) & ... & CP(C_1,T_m) \\ CP(C_2,T_1) & CP(C_2,T_2) & CP(C_2,T_3) & ... & CP(C_2,T_m) \\ CP(C_3,T_1) & CP(C_3,T_2) & CP(C_3,T_3) & ... & CP(C_3,T_m) \\ ... & ... & ... & ... & ... \\ CP(C_n,T_1) & CP(C_n,T_2) & CP(C_n,T_3) & ... & CP(C_n,T_m) \end{bmatrix}$$

The relevance degree of document D for domain ontology O is defined as follows.

Definition 5 (Relevance Degree). *The relevance degree of document D for domain ontology O is defined as the inner product of a document vector D^t and the concept preference matrix CP_{matrix}, where D^t is a vector composed of a term T_j in a document and its weight w_i.*

$$\begin{aligned} Sim(O,D) &= CP_{matrix} \bullet D^t \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} CP(C_i,T_j) \cdot w_j \end{aligned}$$

3 Experimental Results

3.1 Experimental Settings

Seed Ontology and Web Page Set Construction. The seed ontology for the business domain was provided by Poznañ University of Economics. For each concept, we conducted a Google search for the concept and constructed a list of the top 10 retrieved web pages as the web page set for the CP calculation. After applying preprocessing, we calculated the CP value for each concept.

Test Set Construction and Indexing. For the test set to validate the filtering performance, we crawled 1,409 web pages from the Yahoo Picks [11] web site. As the answer set of the business domain, we take 41 web pages in the Business & Economy Yahoo category, which represents approximately 4% of the overall test set. The normalized term frequency (TF) method was used for indexing.

Evaluation Measure. We used precision, recall, and revised accuracy as measures of the filtering effectiveness. Precision and recall are defined as in equations 5 and 6 and Table 1. In Table 1, $a + c$ is the positive set that is relevant for the given domain ontology, and $b + d$ is the negative set that is not relevant for the domain ontology.

$$Precision = \frac{a}{a+b} \tag{5}$$

$$Recall = \frac{a}{a+c} \tag{6}$$

The accuracy measure corresponds to the proportion of all decisions that were correct, as expressed in equation 7. However, the test set in this paper is

Table 1. Precision and recall for a given pattern

	Answer *yes*	Answer *no*
System extracted *yes*	a	b
System extracted *no*	c	d

composed of 4% business domain web pages (positive set) and 96% non-business domain web pages (negative set). Therefore, the overall accuracy performance will depend mainly on how well the system filters the negative set, and will not effectively reflect the system performance in regard to how well it identifies pages in the positive set. Therefore, to better express the accuracy for the positive and negative sets, we defined the slightly revised accuracy R_Accuracy, as shown in equation 9. The revised accuracy measures the harmonic mean of the correct decision ratio in the positive set and the correct decision ratio in the negative set, and represents the average ratio of the correct decisions in the positive and negative sets.

$$Accuracy = \frac{(a+d)}{(a+b+c+d)} \tag{7}$$

$$P\_Accuracy = \frac{a}{(a+c)}, \quad N\_Accuracy = \frac{d}{(b+d)} \tag{8}$$

$$R\_Accuracy = \frac{2 \cdot P\_Accuracy \cdot N\_Accuracy}{(P\_Accuracy + N\_Accuracy)} \tag{9}$$

3.2 Baseline

To the best of our knowledge, there have been few reports on the performance of document filtering for domain ontologies. One related attempt was the ontology-focused document crawling method of Ehrig [12], which crawls documents relevant to a domain ontology by using the terms and meta-data in a document. Given the lack of previous work on document filtering in this area, we decided to employ the relevance measure used by Ehrig [12], which considers the original list of concepts referenced in a document and their term weights as the baseline. The relevance degree between the domain ontology O and the document D in the test set is estimated using the following equation: here, w_t represents the weight of term t in a document, and $\delta(t, C)$ indicates a function that returns 1 if term t refers to concept C in ontology O, and returns 0 otherwise.

$$Sim_{baseline}(O, D) = \sum_t w_t \cdot \delta(t, C), \ t \in D, \ C \in O \tag{10}$$

3.3 Filtering Performance Comparison

We conducted two types of experiment to test the performance reliability of the proposed method on the test set. First, we compared the filtering results

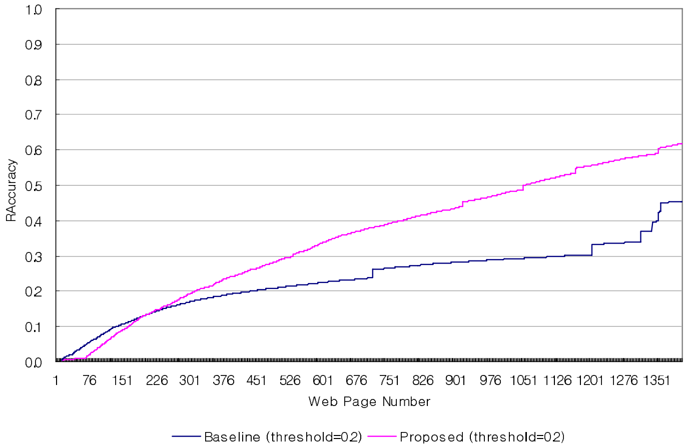


Fig. 2. R_Accuracy of the proposed method and baseline system under optimal threshold value of $threshold = 0.2$

obtained using the $CP_{maxtrix}$ with those obtained using the baseline system on the stream of incoming web pages from the test set (Section 3.3). We then examined the web page rankings generated using the CP_{matrix} and compared them with those obtained using the baseline system (Section 3.3).

Filtering Experiment on Streams of Incoming Web Pages. To examine the effectiveness with which the proposed method filters streams of incoming web pages, we ran a filtering experiment on a sequence of incoming web pages from the test set. The filtering system was applied to the web pages in the sequence in which they came from the test set. A web page was deemed relevant for the domain ontology if it satisfied a certain threshold with varying $threshold \in [0.1, 1.0]$, as expressed in terms of the R_accuracy results. For both methods, the best R_accuracy results are obtained for $threshold = 0.2$.

Figure 2 shows the R_accuracy result obtained using the proposed method and the baseline system with the optimal threshold value of $threshold = 0.2$. The proposed method shows significantly higher R_accuracy values than the baseline system in the overall incoming web page category. For the incoming stream of 1,409 web pages from the test set, the R_accuracy results of the proposed method and the baseline system were 61.74% and 45.41%, respectively.

The above results indicate that the proposed concept preference-based filtering scheme successfully recognizes the preference specification of the domain ontology, suggesting that the scheme represents a good approach to filtering streams of incoming web pages to identify those relevant to a particular domain ontology.

Filtering Experiment for Web Page Ranking. To examine the filtering effectiveness of the proposed concept preference approach in terms of web page

Table 2. Average precision, recall and improvement for the top 100 web pages

	Precision(%)		Recall(%)	
Top N	Baseline	Proposed(Improvement)	Baseline	Proposed(Improvement)
Top 10	14.91	43.58 (192.23)	2.68	4.39 (63.64)
Top 20	14.14	30.67 (116.81)	3.78	5.61 (48.39)
Top 30	13.73	28.37 (106.68)	5.28	8.86 (67.69)
Top 40	12.21	24.58 (101.37)	5.67	9.57 (68.82)
Top 50	11.78	22.75 (93.10)	6.78	11.07 (63.31)

ranking, we carried out a retrieval experiment on the test set for CP_{matrix}, and compared the results with those of the baseline system.

Table 2 lists the average precision, recall, and the performance improvement over the baseline for the top N web pages where N=10, 20,...,50. The data show, for example, that the average precisions of the baseline and proposed systems are 14.91% and 43.58% for the top 10 web pages, and 5.67% and 9.57% for the top 40 web pages, respectively. Thus, the proposed system improved the precision of the baseline by as much as 192.23%, and the recall by as much as 68.82%.

These results indicate that the proposed CP_{matrix} successfully recognizes the preference specification of the domain ontology for a relevant web page, and thus significantly improves the precision and recall performance over the baseline, especially for the top-ranked web pages. The filtering potential of the proposed technique on the top-ranked web pages should facilitate ontology management, because it will allow domain experts to focus on the web pages with the highest rankings.

4 Concluding Remarks

In this work we have developed a document filtering method for domain ontologies based on the concept preference. In a series of experiments on 1,409 web pages crawled from YahooPicks, we found that compared to the baseline method, the concept preference-based technique could more effectively represent the preference specification of the domain ontology for the domain-relevant documents. The proposed method should prove very useful in data-driven applications for domain ontologies such as domain ontology construction, expansion, and evolution. We are currently seeking to develop filtering approaches that consider not only single concepts but also the relations connecting pairs of concepts in the ontology.

Acknowledgments

This study was supported by a grant of National Project for Information Technology Advancement, Ministry of Information and Communication, and the Interoperable EHR Research and Development Center(A050909), Ministry of Health & Welfare, Republic of Korea.

References

1. S. Decker, M. Erdmann, D. Fensel, and R. Studer, "Ontobroker: Ontology based access to distributed and semi-structured information," In R. Meersman et al. (eds.): Semantic Issues in Multimedia Systems, Kluwer Academic Publisher, pp351-369, 1999.
2. H.M. Haav, Learning ontologies for domain-specific information retrieval, W. Abramowicz (Ed), Knowledge-Based Information Retrieval and Filtering from the Web, Kluwer Academic Publishers, 2003, ch 14.
3. F. Abbattista, A. Paradiso, G. Semeraro, F. Zambetta, "An agent that learns to support users of aWeb site," Appl. Soft Comput., Vol. 4, No. 1, pp. 1–12, 2004.
4. S.E. Middleton, N.R. Shadbolt, D.C. De Roure, "Ontological user profiling in recommender systems," ACM Trans. Inform.Syst. (TOIS), Vol. 22, No. 1, pp. 54–88, 2004.
5. B. Sarwar, G. Karypis, J. Konstan, J. Riedl, "Analysis of recommendation algorithms for e-commerce," in: Proceedings of the 2nd ACM Conference on Electronic Commerce, 2000.
6. B. Sarwar, J. Konstan, A. Borchers, J. Herlocker, B. Miller, J. Riedl, "Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system," in: Proceedings of the 1998 Conference on Computer Supported Cooperative Work, 1998.
7. M. Balabanoic, "An adaptive web page recommendation service," in: Proceedings of the First International Conference on Autonomous Agents, pp. 378–385, 1997.
8. S. Singh, P. Dhanalakshmi, L. Dey, "Rough-fuzzy document grading system for customized text information retrieval," Information Processing and Management, Vol. 41, pp. 195–216, 2005.
9. M.W. Berry, Survey of text mining: clustering, classification, and retrieval, Springer, pp.25–42, 2003.
10. J. Pustejovsky, The Generative Lexicon, MIT Press, Cambridge, Massachusetts, 1995.
11. This is available from YahooPicks Online [http://picks.yahoo.com], 2000.
12. M. Ehrig and A. Maedche, "Ontology-focused crawling of web documents," in: Proceedings of ACM Symposium on Applied Computing, 2003.

Qualitative Spatial Relation Database for Semantic Web

Sheng-sheng Wang and Da-you Liu

Key Laboratory of Symbolic Computing and Knowledge Engineering of Ministry of Educaion,
College of Computer Science and Technology, Jilin University,130012, Changchun, China
dyliu@jlu.edu.cn, wss@jlu.edu.cn

Abstract. Geospatial Semantic Web (GSW) has become one of the most prominent research themes in geographic information science over the last few years. The traditional spatial database stores the quantitative data such as coordinate, while GSW needs much qualitative information such as spatial relation. The previous qualitative spatio-temporal systems were all prototype systems which did not support general spatio-temporal relation model and data input. We design the qualitative spatial relation database (QSRDB) based on spatial reasoning. GML data can be converted to QSRDB as data input. OWL ontologies can be generated from QSRDB and applied to GSW systems.

1 Introduction

With the development of Semantic Web and Geographical Information System, the concept of Geospatial Semantic Web (GSW) has been put forward [1]. Within GSW, the qualitative spatial information (such as qualitative spatial relation) is more important than quantitative spatial information.

The traditional spatial database store the quantitative data such as coordinate. But the qualitative information is more close to human thought. Some pure qualitative spatial systems were designed before, such as Place-base GIS proposed by NCGIA[2,3]. For example, in a way finding application, the route instructions (such as "turn right at the Museum") is better than a map with marked routes for understanding.

But the previous qualitative spatial systems can hardly be applied to GSW, because the data input problem has not been solved. Most of spatial data is quantitative, so we need a bridge to convert quantitative spatial data into qualitative one. Another problem is the variety of spatial relation models. There are over 30 spatial relation models, it is a hard work to put them together. So the general spatial relation model is required.

We design the qualitative spatial relation database (or QSRDB for short). The theory of QSRDB is derived from spatial reasoning. Spatial reasoning (SR), the researching field aiming at spatial and/or temporal questions, has widely variety of potential applications in Artificial Intelligence (such as spatial information systems, robot navigation, natural language processing, visual languages, qualitative simulation of physical processes and common-sense reasoning) and other fields (such as GIS and CAD)[4,5].

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 387 – 399, 2006.
© Springer-Verlag Berlin Heidelberg 2006

QSRDB stores the objects and their qualitative relations instead of coordinates. Objects are recorded as identifier and properties. Qualitative relation models are formalized by general framework. Relations of objects are stored by relational DBMS. QSRDB is built from GML data, the standard spatial data format. Technical details will be discussed in the following sections.

2 Spatial Relation Model

Spatial relation is one of the most important theory problems in the fields of spatial reasoning, Geographical Information System (GIS) and computer vision, as important as the spatial object itself. Spatial relation plays an important role in the process of spatial reasoning, spatial query, spatial analysis, spatial data modeling and map interpretation. Spatial relation is the relation between the objects with spatial characters. It usually consists of topological relations, metric relations and order relations. These relations are the bases of spatial data organizing, querying, analyzing and reasoning.

Qualitative spatial relations are more important than the quantitative one, since they are more close to human thought. Topological relation between spatial objects is the most elementary qualitative relation in space. It is the basic aspect of spatial reasoning. The research of topological relation plays an important role in spatial reasoning. There are also many researches focused on other spatial relations (such as direction, distance, size, shape etc.) and temporal relations. Nowadays, the studies of spatial relations and the combination of single spatial relations increase quickly.

2.1 General Framework

Many works dedicated to spatial relation model are independent, although all sorts of spatial relations are relational more or less. That will undoubtedly make obstacle to build QSRDB. So we investigated the general spatial relation framework. Furthermore, current SR works rarely consider the implement cost, such as time complex for judging spatial relation. Our framework is based on basic operations in spatial database which is easy to carry out and requires lower space and time.

Our spatial relation framework contains three parts:

(1) Definition of Object

One object is defined by the atomic parts. For different relation models, the definition may be different.

In this paper, object is denoted by uppercase. The atomic parts are denoted by lowercase and they are the properties of the object.

For instance, X has two atomic parts: X.x1 and X.x2.

(2) Definition of Relations

In this paper, all relations are binary, in another word they are between two objects. The basic relations of a model are called JEPD(Jointly Exhaustive and Pairwise Disjoint) relations. They are defined by atomic function between atomic parts of two objects. Atomic function makes all the spatial relation models have a unique formal method. So it is easy to define and maintain all the models. The atomic functions are using the basic operations of the platform such as MapInfo or ArcInfo.

In this paper, we use MapInfo products. There are only two atomic functions in our system:

```
//p1,p2 are multi-dimensional geometries such as point, poly line and region.
Bool Contains (p1,p2);  //If p1 contains p2 return true, else return false.
Bool Intersects (p1,p2);  //If p1 Intersects p2 return true, else return false.
```

The two functions could be executed by MapBasic 5.0 and MapInfo desktop 7.0 or above versions. Each relation model has a function JudgeRel(X,Y) , it defines the basic relation by atomic functions operations.

Two special basic relations named SAME and DEFALUT are defined for saving storage space.

R(X,X) = SAME or R(X,Y) = SAME when X=Y .

DEFALUT is used to reduce the storage space. DEFALUT relation dose not save in dataset.

R(X,Y) could not be found in relation dataset   R(X,Y) = DEFALUT.

For example, in RCC-8 model we set DEFALUT = DC and this will reduce 90% storage space in most applications. This will also save the retrieval time.

Another method to save space is using *reverse* function.

If $R_1 = R(A,B)$ and $R_2 = R(B,A)$ then R_2=*Reverse*(R_1) , R_1=*Reverse*(R_2)

In relation dataset, only half relation is actually stored, the others is obtained by *Reverse*(R), so we can further reduce 50% space.

Function AddRel and GetRel implement the above strategies.

Algorithm 1. Add relation to dataset

```
AddRel( X, Y, R)
{
    if R=DEFAULT return;
    if X >Y then
   {
    $X \leftrightarrow Y$;
    $R \leftarrow$ Reverse (R);
      }
      Add (X, Y, R) to dataset;
}
```

Algorithm 2. Get relation from dataset

```
GetRel( X, Y )
{
    if X > Y then
   {
    $X \leftrightarrow Y$;
    rev $\leftarrow$ true;
      }
      if  get (X, Y, R) in dataset is failed  then R $\leftarrow$ DEFAULT;
      if rev then R $\leftarrow$Reverse(R);
      return R;
}
```

(3) Definition of Reasoning

Only one type of spatial reasoning is involved in this paper: the composition table. The most prevalent form of qualitative reasoning is based on the composition table. A compositional inference is a deduction, it decides R_1(a,c) from R_2(a,b) and R_3(b,c). The validity of compositional inferences does not depend in many cases on the constants involved but only on the logical properties of the relations. In such case the composition of pairs of relations can be maintained for table looking up as and when required. This technique is of particular significance when we are dealing with relational information involving a fixed set of relations. Given a set of n JEPD relations, one can build a n×n composition table the relationships between x and z for a pair of relations R_1(x,y) and R_2(y,z). In general, each entry will be a disjunction because of the qualitative nature of the calculus.

In this paper, the composition relation is formally defined as:

Definition 1 （Composition Relation）**.** Let R_1 and R_2 be two relations, the Composition Relation, between R_1 and R_2,is defined as follows:

$$R_1 \, \mathrm{o} R_2 = \{ \, r \,|\!.\, A,B,C \; [A \; R_1 \; B \text{ and } B \; R_2 \; C \text{ and } A \; r \; C] \, \}$$

Bennett[6] and Duntsch[7] pointed out that the definition of composition table in qualitative spatial reasoning is a weak composition. The composition relation within the relational algebra is defined by the following:

$$(\forall x, y, z)[xRz \wedge zRy \Rightarrow xT_0 y \vee ... \vee xT_k y] \tag{1}$$

$$(\forall x, y)[xT_i y \Rightarrow (\exists z) xRz \wedge zRy] \tag{2}$$

The composition relation of qualitative spatial reasoning uses only a constraint condition among them.

Composition table is used to reduce geometrical calculation while building QSRDB, function ***Composition*** return composition result of two relations:

$$Composion(R_1, R_2) = \begin{cases} R_1 \circ R_2 & \text{if } |R_1 \circ R_2| = 1 \\ NULL & \text{if } |R_1 \circ R_2| > 1 \end{cases} \tag{3}$$

NULL means composition of R1 and R2 could not be unique determined.

For example, in RCC model,

TPP oEC = {DC,EC}   Composion(TPP, EC) = NULL.

If Composion(R(X,Y) , R(Y,Z)) ≠NULL, we need not calculate R(X,Z).

2.2 Some Spatial Relation Models

Most current spatial relation models could be formalized by the above framework. Here we take some popular models for examples.

(1) RCC Models

The best known topological theory is Region Connection Calculus (RCC for short) [8]. It is a mereo-topological theory based on spatial region ontology. Many spatial relation models (such as RCC-5, RCC-7 , RCC-8 , RCC-10, RCC-13) are deduced by RCC theory. Among them, RCC-8 is well-known in state-of-the-art Geographical Information System, spatial database, visual languages and other applied fields. It has one primitive dyadic relation *C(x,y)* which means "x is connected with y". Eight JEPD relations is deduced by *C(x,y)*.

$(D1)\ DC(x,y) \equiv def.\ \neg C(x,y)$
$(D2)\ P(x,y) \equiv def.\ \forall z[C(z,x) \rightarrow C(z,y)]$
$(D3)\ EQ(x,y) \equiv def.\ P(x,y)\ \&\ P(y,x)$
$(D4)\ O(x,y) \equiv def.\ \exists z[P(z,x)\ \&\ P(z,y)]$
$(D5)\ DR(x,y) \equiv def.\ \neg O(x,y)$
$(D6)\ PO(x,y) \equiv def.\ O(x,y)\ \&\ \neg P(x,y)\ \&\ \neg P(y,x)$
$(D7)\ EC(x,y) \equiv def.\ C(x,y)\ \&\ \neg O(x,y)$
$(D8)\ PP(x,y) \equiv def.\ P(x,y)\ \&\ \neg P(y,x)$
$(D9)\ TPP(x,y) \equiv def.\ PP(x,y)\ \&\ \exists z[EC(z,x)\ \&\ EC(z,y)]$
$(D10)\ NTPP(x,y) \equiv def.\ PP(x,y)\ \&\ \neg \exists z[EC(z,x))\ \&\ EC(z,y)]$
$(D11)\ PI(x,y) \equiv def.\ P(y,x)$
$(D12)\ PPI(x,y) \equiv def.\ PP(y,x)$
$(D13)\ TPPI(x,y) \equiv def.\ TPP(y,x)$
$(D14)\ NTPPI(x,y) \equiv def.\ NTPP(y,x)$

{DC,EC,PO,TPP,NTPP,TPPI,NTPPI,EQ} are JEPD basic relations of RCC-8 (Fig. 1).

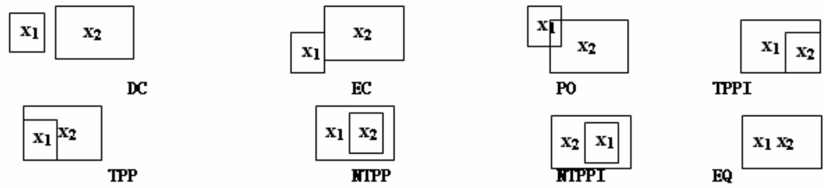


Fig. 1. RCC-8 Basic Relations

Firstly RCC-8 is defined by 1st-order logic and propositional logic, later Bennett encoded the basic relations of it modal logic. **I** is an interior operator which results in the following axioms:

(1) $\mathbf{I}X \rightarrow X$
(2) $\mathbf{II}X \leftrightarrow \mathbf{I}X$
(3) $\mathbf{I}T \leftrightarrow T$ (for any tautology T)
(4) $\mathbf{I}(X \wedge Y) \leftrightarrow \mathbf{I}X \wedge \mathbf{I}Y$

(1) and (2) correspond to the modal logic **T4** and (3)(4) are hold for any **K** system, so **I** is a modal **S4**-operator.That is to say, RCC-8 equal to **S4** modal logic system.

We define RCC-8 by our general framework which is equal to the above formalization, but more convenient to be applied in computer systems.

Firstly, object is defined by two parts: interior and boundary (Fig. 2).

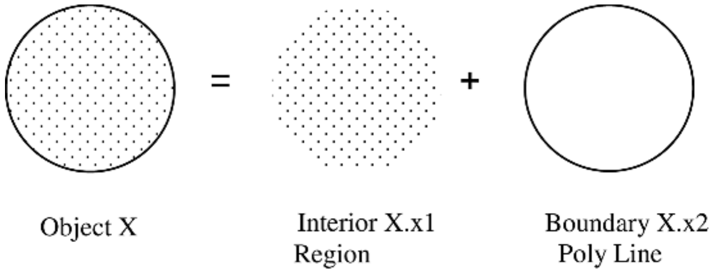


Fig. 2. Define Object for RCC-8 Relation

Secondly, the RCC-8 basic relations are determined by the following algorithm.

Algorithm 3. Calculate RCC-8 basic relations

```
JudgeRel (X,Y)
{
    If Not Intersects(X.x1,Y.y1) then
     {
       If Intersects(X.x2,Y.y2) then return EC; }
       else return DC;
     }
     If Contains(X.x1,Y.y1) then
     {
       If Contains (Y.y1,X.x1) then return EQ;
       If Intersects(X.x2,Y.y2) then return TPPI; }
       else return NTPPI;
     }
     If Contains(Y.y1,X.x1) then
     {
       If Intersects(X.x2,Y.y2) then return TPP; }
       else return NTPP;
     }
     return PO;
}
```

Two special relations of RCC-8: DEFAULT=DC ; SAME=EQ .

Table 1 is the reverse relations of RCC-8.

Table 1. Reverse relations of RCC-8

R	DC	EC	PO	TPP	NTPP	TPPI	NTPPI	EQ
***Reverse*(R)**	DC	EC	PO	TPPI	NTPPI	TPP	NTPP	EQ

Table 2 gives *Composion*(R) of RCC-8.

Table 2. Composion results of RCC-8 basic relations

	DC	EC	PO	TPP	NTPP	TPPI	NTPPI	EQ
DC	NULL	NULL	NULL	NULL	NULL	DC	DC	DC
EC	NULL	NULL	NULL	NULL	NULL	NULL	DC	EC
PO	NULL	NULL	NULL	NULL	NULL	NULL	NULL	PO
TPP	DC	NULL	NULL	NULL	NTPP	NULL	NULL	TPP
NTPP	DC	DC	NULL	NTPP	NTPP	NULL	NULL	NTPP
TPPI	NULL	NULL	NULL	NULL	NULL	NULL	NTPPI	NTPPI
NTPPI	NULL	NULL	NULL	NULL	NULL	NTPPI	NTPPI	NTPPI
EQ	DC	EC	PO	TPP	NTPP	TPPI	NTPPI	EQ

(2) Cardinal Direction

A cardinal direction is a binary relation involving a target object *A* and a reference object *B*, and a symbol that is a non-empty subset of {*NW*, *N*, *NE*, *W*, *O*, *E*, *SW*, *S*, *SE*} whose semantics are motivated by the grid formed around the reference object. Serafino gave the definition of 4-connected direction matrix[9], and mentioned that only 218 4-connected basic direction relations can be realized out of 2^9=512 possible combinations of the nine atomic relations.

For example, in Fig. 3 R(X,Y) = {E,S,SE}.

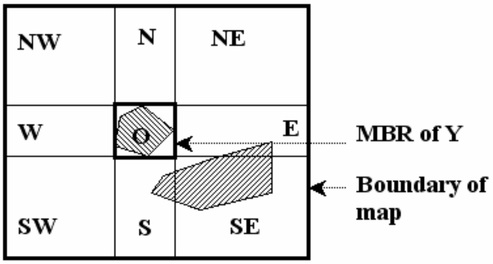


Fig. 3. Nine direction tiles formed around reference object

We can also define cardinal direction by our framework.

Firstly, we use 9 rectangles to build the reference system of Y. Denote $y_1,\ldots, y_9$ for {*NW*, *N*, *NE*, *W*, *O*, *E*, *SW*, *S*, *SE*}.

Boundary of y_5 (O) is MBR (Minimal Boundary Rectangle) of Y. External boundary is boundary of map i.e. all the objects are located within it.

Algorithm 4. Calculate Cardinal Direction basic relations

```
global D(1..9) = { NW, N, NE, W, O, E, SW, S, SE };
JudgeRel (X,Y)
{
 R ←∅;
  for i ← 1 to 9
```

```
    {
            if Intersects ($Y.y_i$, X) then R $\leftarrow$ R $\cup$ D(i);
    }
  return R;
  }
```

SAME=C; DEFAULT=N (or any other relations) ;
Composition table of cardinal direction can be found in [10] .

(3) Broad Boundary

Basing on 9-intersections model, Clementini and Di Felice [11] proposed broad boundary model for uncertain spatial topological relation. The method extends 9-intersection matrix to:

$$\begin{pmatrix} A^{\circ} \cap B^{\circ} & A^{\circ} \cap \Delta B & A^{\circ} \cap B^{-} \\ \Delta A \cap B^{\circ} & \Delta A \cap \Delta B & \Delta A \cap B^{-} \\ A^{-} \cap B^{\circ} & A^{-} \cap \Delta B & A^{-} \cap B^{-} \end{pmatrix}$$

Where $A^{\circ}, \Delta A$ and A^{-} is interior, broad boundary and exterior of uncertain region respectively.

In our framework, $A^{\circ}, \Delta A$ are atomic parts. The algorithm detail is too long to be listed here.

Other models, such as Interval Algebra, Naming Distance … , are all implemented with the framework. In fact, most certain models and region based uncertain models are supported by the framework.

3 Qualitative Spatial Relation Database

Relation dataset of QSRDB is stored in MS SQL Server. Major part of relation dataset has three fields:

Object$_1$	Object$_2$	Relation

Strategies for optimizing storage space has been discussed in section 2.1 .

Since spatial data are multi-source and isomerous, we take GML as quantitative data input standard for QSRDB. The Geography Markup Language (GML) [12] is an XML language created under the auspices of the Open GIS Consortium, whose mission statement is to facilitate the “full integration of geospatial data and geoprocessing resources into mainstream computing and to foster the widespread use of inter operable geoprocessing software and geodata products throughout the information infrastructure”. GML, which could be considered the flagship effort in geoprocesssing, is a multi-stage effort that has reached version 3.10. By designing the language in multiple stages, the standardization body wants to make sure that the language evolves naturally and incorporates more and more features over time. GML’s definition is based on XML Schema and tries to take advantage of its full feature set. GML is able to describe a wide variety of geographical objects by combining its built-in data types.

Being an application of XML, GML is designed to take advantage of the XML standards framework; documents can be readily transformed into a variety of presentation formats, most notably vector formats and raster graphics such as maps. Version 3.0 has added features for temporal GIS, including time stamps, events, and histories, as well as units of measure and the possibility of grouping features into layers, to name but a few. But since GML is used in practice and competing standards converge with it, many users consider it mature at its current stage and take advantage of its features. For researchers, the advantages of using GML include the availability of test data sets and the incentive to use the modeling knowledge of the geographic data management community in their prototypes, which can in turn be more flexibly leveraged by other researchers and industrial partners alike. Due to the strict semantics of GML, there is also the potential to benefit from the domain modeling that is part of the standard.

The follow algorithm builds spatial relations based on the objects extracted from GML.

Algorithm 5. Building QSRDB

```
Input: N objects
Output: Relation dataset
Build()
{
    for I ← 1 to N
    for J ← 1 to N
    AddRel(I, J, JudgeRel(I,J) );
}
```

Base on composition reasoning of spatial reasoning, algorithm Build() can be optimized.

Algorithm 6. Building QSRDB base on spatial reasoning

```
Input: N objects
Output: Relation dataset
STRBuild()
{
    for I ← 1 to N
      AddRel(I, I, SAME );
    for I ← 1 to N
    for J ← 1 to N
      add (I, J) to UC; // UC is the collection of all uncalculated pairs
    for ∃(A,B)∈UC
    {
      R1 ← JudgeRel(A,B);
      Update(A, B, R1);
      Update(A, B, reverse(R1) );
    }
}

Update(A, B, R)
```

```
{
    AddRel(A, B, R); //add R(A,B) to relation dataset
    Delete (A, B) in UC;
    for $1 \le C \le N$ and $C \ne A$, B and (B,C) $\notin$ UC and (A,C) $\in$ UC
    {
            $R_2 \leftarrow$ GetRel (B,C); // get R(B,C)
      $R_3 \leftarrow$ Combosition ($R_1$, $R_2$);
      if $R_3 \ne$ NULL then
      {
          Update(A,C, $R_3$);
          Update(C,A, reverse($R_3$) );
      }
    }
}
```

4 Applied to Geospatial Semantic Web

Recently, the notion and concept of ontologies have gained increased attention among researchers in geographic information science to address the many problems of geographic data dealing with spatial data exchange and representation. Ontologies are essential to the creation and use of data exchange standards, and to the design of human-computer interfaces. Ontologies are also important in the solution of problems of heterogeneity and interoperability of geographic data.

The spatial relations are important properties in spatial ontologies. In Kolas's architecture of ontologies, the "Common Language for Geospatial Relationships" is the core parts of "Geospatial Filter Ontology" [13]. Our work in this paper will help improving Kolas's architecture.

The "general qualitative spatial relation framework" is a good common language for geospatial relationships which has better compatibility and easier to implement. The converting method from GML to QSRDB can solve the data input problem. Fig. 4 shows the improved architecture of GSW.

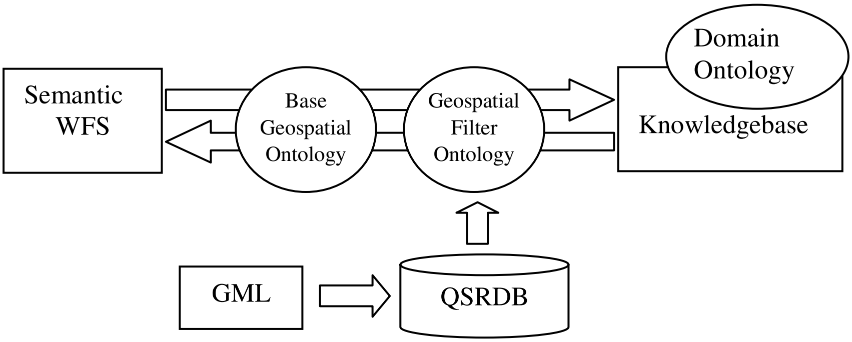


Fig. 4. Improved architecture of GSW

GML documents can be imported to QSRDB by the method discussed before, and QSRDB can be translated to OWL ontologies by the following steps:

Step 1: Collect all the concepts related to spatial objects in QSRDB
Step 2: Classify the collected concepts.
Step 3: Define the relationship between concepts. Most concepts are possible to be defined by 'subClassOf' relationship.
Step 4: Define the relationship by the general framework of spatial relations.
Step 5: Build the web ontologies using the OWL language.ï

We use china map for experiment (Fig. 5).

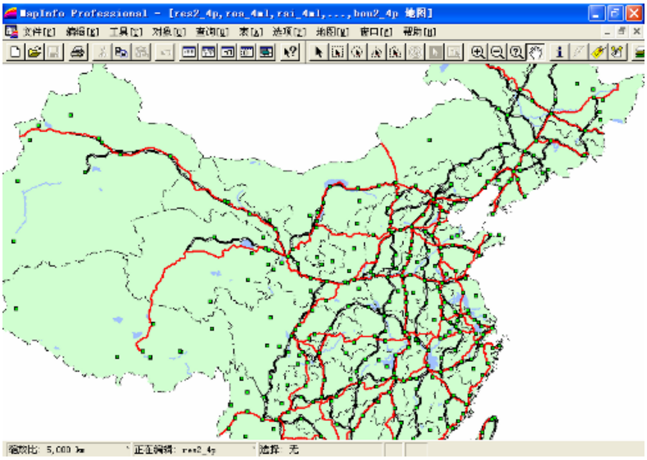

MapInfo Professional - [res2_4p, roa_4ml, rai_4ml, ..., bou2_4p 地图]
文件[F] 编辑[E] 工具[T] 对象[O] 查询[Q] 表[A] 选项[O] 地图[M] 窗口[W] 帮助[H]
缩放比: 5,000 km
正在编辑: res2_4p
选择: 无

Fig. 5. The China Map Project by MapInfo

There are 5 layers in the China project, totally 1531 objects.

Table 3. Objects in China Map Project

Layer	Objects
province	34
water	826
railway	189
resource	331
road	151
Total	**1531**

We applied RCC-8, Cardinal Direction and Naming Distance spatial relations in this project.

The OWL ontologies generated from the China project is:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"

   <owl:Class rdf:ID="China">
   <rdfs:subClassOf rdf:resource=" http://www.w3.org/2002/07/owl#Thing"/>
   </owl:Class>
   <owl:Class rdf:ID="Beijing">
   <rdfs:subClassOf rdf:resource="#China"/>
   <owl:DC rdf:resource="#Jilin "/>
   ...
   ...
   <owl:NE rdf:resource="#Jilin "/>
   ...
   ...
   <owl:FAR rdf:resource="#Jilin "/>
   ...
   ...
```

QSTDB can also be applied in the following fields:

(1) Qualitative Spatio-temporal Query

Qualitative spatio-temporal query excels metric spatio- temporal query in time cost and result understandability. Qualitative query language can easily be built base on QSTDB.

(2) Spatio-temporal Data Mining

Since researchers do not want to mine the patterns from the coordinates but the spatio-temporal relations. So QSTDB is quite suitable to spatio-temporal data mining.

(3) Way Finding

Route instructions are often used in way finding systems, because they are better than map both in understanding and transmission. The relation of a navigator's location to the location of a waypoint can be described qualitatively. Topology and distance relations can be use in way finding systems.

5 Conclusion

In this paper, we design the qualitative spatial relation database (QSRDB) based on spatial reasoning. The general spatial relation framework is put forward. It uses atomic operations of GIS platform, so it is easy to implement. QSRDB use the general spatial data standard GML for data input. It is compatible to most spatial systems. By using spatial reasoning technology, QSRDB is optimized both on reducing storage space and process time. GML data can be converted to QSRDB as data input. OWL ontologies can be generated from QSRDB and applied to GSW systems. Compared with previous qualitative spatio-temporal systems such as PB-GIS [1,2], QSTDB is a practical system rather than a theory prototype . QSTDB could also be applied to other fields such as qualitative spatio-temporal query, spatio-temporal data mining and way finding systems.

Acknowledgments

Supported by NSFC Major Research Program 60496321, Basic Theory and Core Techniques of Non Canonical Knowledge; National Natural Science Foundation of China under Grant Nos. 60373098, 60573073, the National High-Tech Research and Development Plan of China under Grant No.2003AA118020, the Major Program of Science and Technology Development Plan of Jilin Province under Grant No. 20020303, the Science and Technology Development Plan of Jilin Province under Grant No. 20030523., Youth Foundation of Jilin University (419070100102).

References

1. Egenhofer, M.J. Toward the Semantic Geospatial Web. In Proceedings of the Tenth ACM International Symposium on Advances in Geographic Information Systems, McLean, Virginia, 2002.
2. KEMP K K,GOODCHILD M F ,MARK D M,et al . Varenius :NCGIA's project to advance geographic information science [A] . Proceedings of Geographical Informationp97 :from Research to Applications through Cooperation[C] . Amsterdam, 1997. 25 - 31.
3. MARK D M, FREKSA C ,HIRTLE S ,et al . Cognitive models of geographical space [J] . International Journal of Geographical Information Science ,1999 ,13 (8) :747 - 774.
4. A.G.Cohn and S.M. Hazarika, Qualitative Spatial Representation and Reasoning: An Overview. Fundamental Informatics, 2001,46(1-2):1~29.
5. M.Teresa Escrig ,Francisco Toledo. Qualitative Spatial Reasoning: Theory and Practice. Ohmsha published,1999
6. B. Bennett, A.G. Cohn, A. Isli, Combining multiple representations in a spatial reasoning system, in: Proc. 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-97), Newport Beach, CA,1997. 37-45
7. I. Duntsch, H. Wang, S. McCloskey, Relation algebras in spatial reasoning, in: E. Or lowska, A. Szatas(Eds.), Extended Abstracts of the 4th Seminar on Relational Methods in Algebra, Logic, and Computer Science, 1998, pp. 63–68.
8. D. Randell, Z. Cui, and A. Cohn. A spatial logic based on regions and connection. In Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning, pages 165--176. Morgan Kaufmann, 1992.
9. Serafino C. and Paolino D. F. "Cardinal Directions between Spatial Objects: the Pairwise-Consistency Problem", Information Sciences, vol 164 pp.165-188, 2004.
10. Skiadopoulos S. and Koubarakis M. "Composing Cardinal Direction Relations", SSTD LNCS 2121, Springer-verlag, Berlin, pp. 299-317, 2001
11. Clementini, E., Di Felice, P. An algebraic model for spatial objects with indeterminate boundaries. In: Burrough, P.A., Frank, A.U. eds. Geographic Objects with Indeterminate Boundaries. London: Taylor & Francis, 1996. 155~169.
12. http://www.opengis.net/gml/, 2004.
13. Dave Kolas, John Hebeler, and Mike Dean. Geospatial Semantic Web: Architecture of Ontologies. M.A. Rodríguez et al. (Eds.): GeoS 2005, LNCS 3799, pp. 183 – 194, 2005.

Automatic Creation and Simplified Querying of Semantic Web Content: An Approach Based on Information-Extraction Ontologies

Yihong Ding[1,⋆], David W. Embley[1,⋆], and Stephen W. Liddle[2,⋆⋆]

[1] Department of Computer Science,
[2] Information Systems Department,
Brigham Young Univeristy, Provo, Utah 84602, U.S.A
{ding, embley}@cs.byu.edu,
liddle@byu.edu

Abstract. The semantic web represents a major advance in web utility, but it is currently difficult to create semantic-web content because pages must be semantically annotated through processes that are mostly manual and require a high degree of engineering skill. Furthermore, users need an effective way to query the semantic web, but any burden placed on users to learn a query language is unlikely to garner sufficient user support and interest. Unfortunately, both the creation and use of semantic-web pages are difficult, and these are precisely the processes that must be made simple in order for the semantic web to truly succeed. We propose using information-extraction ontologies to handle both of these challenges. In this paper we show how a successful ontology-based data-extraction technique can (1) automatically generate semantic annotations for ordinary web pages, and (2) support free-form, textual queries that will be relatively simple for end users to write.

1 Introduction

The sheer volume of web content forces people to rely on machines to help search for information. Search engines help, but by themselves are not enough. Search engines do a good job ranking billions of web pages and identifying useful candidates, often presenting the page a user wants within the first few search results. The problem, however, is not what search engines *do*, but what they *cannot do*. Keyword-based searching restricts the types of questions people can ask. For example, users cannot make requests like, "Find me a red Nissan for under $5000 – it should be a 1990 or newer and have less than 120K miles on it." The required information is out there on the web, but traditional search engines cannot answer this type of request because they do not know how to match the specified concepts in the request to data instances on the web.

[⋆] Supported by NSF grant #0414644.

[⋆⋆] Supported by the Kevin and Debra Rollins Center for eBusiness at Brigham Young University under grant EB-05046.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 400–414, 2006.

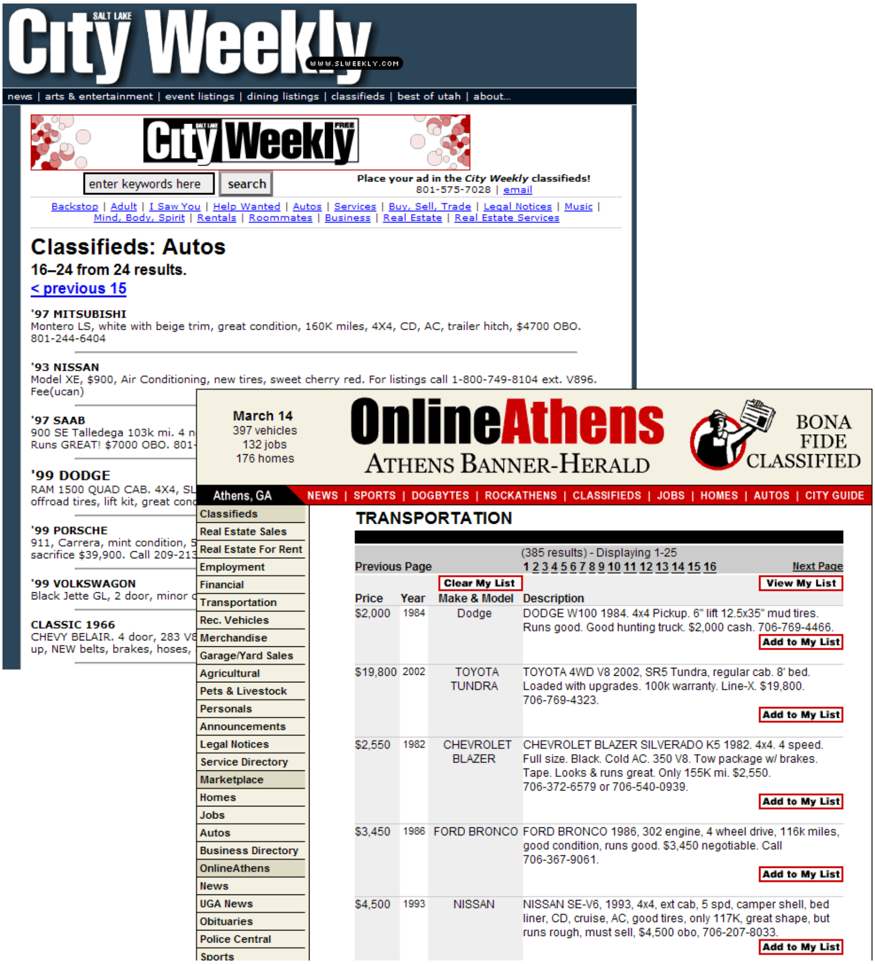
© Springer-Verlag Berlin Heidelberg 2006

A solution to this problem is to design a new type of machine-understandable web representation and develop web pages based on the new format, or in other words develop the *semantic web* [2]. Semantic-web proponents propose making web content machine understandable through the use of *ontologies*, which are commonly shared, explicitly defined, generic conceptualizations [7]. But then one of the immediate problems we face is how to deal with current web pages. There are billions of pages on the current web, and it is impractical to ask web developers to rewrite their pages according to some new, semantic-web standard, especially if this would require tedious manual labeling of documents.

Web semantic annotation research attempts to resolve this problem. The goal of web semantic annotation is to add comments to web content so that it becomes machine understandable. Unlike an annotation in the normal sense, which is an unrestricted note, a semantic annotation must be explicit, formal, and unambiguous: *explicit* makes a semantic annotation publicly accessible, *formal* makes a semantic annotation publicly agreeable, and *unambiguous* makes a semantic annotation publicly identifiable. These three properties enable machine understanding, and annotating with respect to an ontology makes this possible. In this paper we show how to automatically annotate existing data-rich web pages with respect to an ontology.

To clarify our intentions, we give an example. Figure 1 shows two ordinary, human-readable web pages for selling cars. Our system can annotate these pages automatically with respect to a given ontology about car advertisements and thus can convert them to semantic-web pages so that these pages also exist in machine-readable form. We store these annotations in such a way that we can directly query them using an available semantic-web query language (SPARQL [15] for our particular implementation). This entire process allows us to query the content of web pages not originally designed for the semantic web, thus, a request equivalent to “Find me a red Nissan for under $5000 – it should be a 1990 or newer and have less than 120K miles on it” over the pages in Figure 1 would yield results such as those in Figure 2. The results in Figure 2 are *actual answers* to the query in a table whose header attributes are the concept names from the given car-ads ontology, restricted to those concepts mentioned in the query. In addition, there is always one additional attribute, *Source*, whose values are links back into the original documents at the location where the information is provided. When a user clicks on *Car1* (the link in the first row in Figure 2), for example, the document in Figure 1 from the Athens site appears, except it would be scrolled to the right place and the information requested in the query would be highlighted.

Our automated semantic annotation approach employs a unique ontology-based data recognizer that uses information-extraction (IE) ontologies. A unique characteristic of this approach is the use of instance recognition semantics inside ontologies to help specify annotation domains and perform data recognition. Our approach solves a common annotation problem of requiring “a set of heuristics for post-processing and mapping of the IE results to an ontology” [9].



SALT LAKE City Weekly
www.slweekly.com
news | arts & entertainment | event listings | dining listings | classifieds | best of utah | about...
City Weekly FREE
enter keywords here
search

Place your ad in the *City Weekly* classifieds!
801-575-7028 | email

Backstop | Adult | I Saw You | Help Wanted | Autos | Services | Buy, Sell, Trade | Legal Notices | Music | Mind, Body, Spirit | Rentals | Roommates | Business | Real Estate | Real Estate Services

Classifieds: Autos

16–24 from 24 results.

< previous 15

'97 MITSUBISHI
Montero LS, white with beige trim, great condition, 160K miles, 4X4, CD, AC, trailer hitch, $4700 OBO.
801-244-6404

'93 NISSAN
Model XE, $900, Air Conditioning, new tires, sweet cherry red. For listings call 1-800-749-8104 ext. V896.
Fee(ucan)

'97 SAAB
900 SE Talledega 103k mi. 4 n
Runs GREAT! $7000 OBO. 801-

'99 DODGE
RAM 1500 QUAD CAB. 4X4, SL
offroad tires, lift kit, great cond

'99 PORSCHE
911, Carrera, mint condition, 5
sacrifice $39,900. Call 209-213

'99 VOLKSWAGON
Black Jette GL, 2 door, minor c

CLASSIC 1966
CHEVY BELAIR. 4 door, 283 V8
up, NEW belts, brakes, hoses,

March 14
397 vehicles
132 jobs
176 homes

TRANSPORTATION

(385 results) - Displaying 1-25

Price	Year	Make & Model	Description
$2,000	1984	Dodge	DODGE W100 1984. 4x4 Pickup. 6" lift 12.5x35" mud tires. Runs good. Good hunting truck. $2,000 cash. 706-769-4466. Add to My List
$19,800	2002	TOYOTA TUNDRA	TOYOTA 4WD V8 2002, SR5 Tundra, regular cab. 8' bed. Loaded with upgrades. 100k warranty. Line-X. $19,800. 706-769-4323. Add to My List
$2,550	1982	CHEVROLET BLAZER	CHEVROLET BLAZER SILVERADO K5 1982. 4x4. 4 speed. Full size. Black. Cold AC. 350 V8. Tow package w/ brakes. Tape. Looks & runs great. Only 155K mi. $2,550. 706-372-6579 or 706-540-0939. Add to My List
$3,450	1986	FORD BRONCO	FORD BRONCO 1986, 302 engine, 4 wheel drive, 116k miles, good condition, runs good. $3,450 negotiable. Call 706-367-9061. Add to My List
$4,500	1993	NISSAN	NISSAN SE-V6, 1993, 4x4, ext cab, 5 spd, camper shell, bed liner, CD, cruise, AC, good tires, only 117K, great shape, but runs rough, must sell, $4,500 obo, 706-207-8033. Add to My List

Fig. 1. Sample Car Ads from Salt Lake City Weekly and Athens Banner-Herald Sites

We give the details[1] of our contribution of automatically creating semantic-web content so that we can directly query it as follows. Section 2 describes information-extraction ontologies, which are the basis for our automated semantic-web annotation tool. Section 3 describes our prototype work on automatically annotating existing web pages so that they can be used for the semantic web, and Section 4 shows how we can directly query pages annotated for the semantic web. Section 5 provides experimental evidence about the accuracy of our annotation system as well as pragmatic consideration. We conclude in Section 6.

[1] Since this paper gives a full, broad vision of our approach to both the creation and use of semantic-web pages, our presentation is necessarily high level. We provide as much detail as space allows and refer the interested reader to additional papers that augment ideas and results presented here.

Color	Make	Price	Year	Mileage	Source
	Nissan	$4,500	1993	117,000	Car1
...	...	...	...	...	...
red	Nissan	$900	1993		Car13
...	...	...	...	...	...

Fig. 2. Query Results

2 Ontologies for Semantic Annotation

In semantic-web applications, ontologies describe formal semantics for applications, and thus make information sharable and machine-understandable. The work of semantic annotation is, however, more than just knowledge representation. Semantic annotation applications must also establish mappings between ontology concepts and data instances within documents so that these data instances become sharable and machine-understandable. In this section, we introduce information-extraction ontologies and show that they are useful both for representing knowledge and for establishing mappings between ontology concepts and document data instances.

2.1 Information Extraction Ontologies

We have described information-extraction ontologies elsewhere [6], but to make our paper self-contained, we briefly reintroduce them here.[2] An *extraction ontology* specifies named sets of objects, which we call *object sets* or *concepts*, and named sets of relationships among object sets, which we call *relationship sets.* Figure 3 shows a graphical rendition of an extraction ontology for car advertisements. The extraction ontology has two types of concepts: lexical concepts (enclosed in dashed rectangles) and nonlexical concepts (enclosed in solid rectangles). A concept is *lexical* if its instances are indistinguishable from their representations. *Mileage* is an example of a lexical concept because its instances (e.g. "117K" and "5,700") represent themselves. A concept is *nonlexical* if its instances are object identifiers, which represent real-world objects. *Car* is an example of a nonlexical concept because its instances are identifiers such as, say, "Car1", which represents a particular car in the real world. An extraction ontology also provides for explicit concept instances (denoted as large black dots). We designate the main concept in an extraction ontology by marking it with "->●" in the upper right corner, which denotes that the object set *Car* becomes ("->") an object instance ("●") for a single car ad.

Figure 3 also shows relationship sets among concepts, represented by connecting lines, such as the connecting line between *Car* and *Year.* The numbers near the connections between relationship sets and object sets are participation constraints. Participation constraints give the minimum and maximum participation of an object in an object set with respect to the connected relationship

[2] We mention, in passing, that the ontological basis for our extraction ontologies has been fully formalized in terms of predicate calculus. (See Appendix A of [5].)

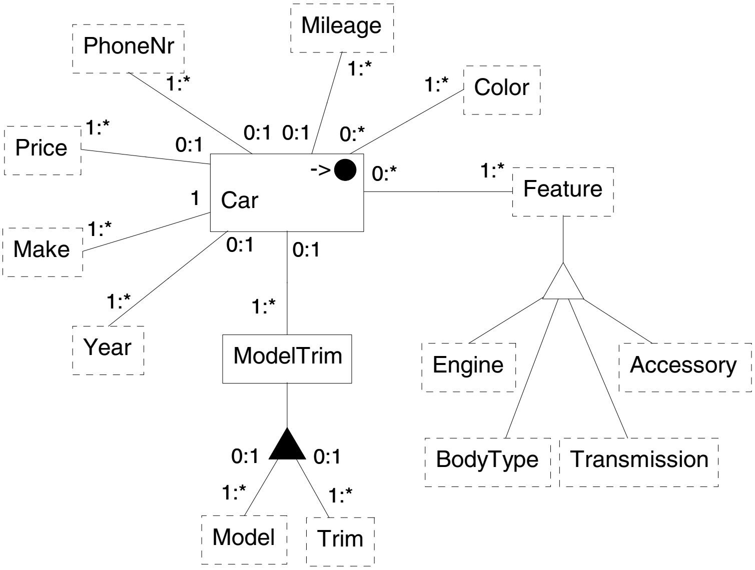


Fig. 3. Graphical Component of an Extraction Ontology

set. For example, the *0:1* participation constraint on *Car* in the *Car-Mileage* relationship set denotes that a car need not have a mileage in a car ad, but if it does, it has only one. A white triangle defines a generalization/specialization relationship, with the generalization concept connected to the apex of the triangle and one or more specialization concepts connected to its base. In Figure 3, for example, *Feature* is a generalization of *Engine* and *BodyType*, among other concepts. The white triangle can, of course, appear repeatedly, and thus we can have large *ISA* hierarchies in an extraction ontology. A black triangle defines an aggregation with the super-part concept connected to the apex of the triangle and the component-part concepts connected to its base. In Figure 3, for example, *ModelTrim* is an aggregation of *Model* and *Trim*. Like *ISA* hierarchies, large *PartOf* hierarchies are also possible.

As a key feature of extraction ontologies, the concepts each have an associated data frame. A *data frame* describes information about a concept—its external and internal representations, its contextual keywords or phrases that may indicate the presence of an instance of the concept, operations that convert between internal and external representations, and other manipulation operations that can apply to instances of the concept along with contextual keywords or phrases that indicate the applicability of an operation. Figure 4 shows sample (partial) data frames for the concepts *Price* and *Make* in our ontology for car advertisements. As Figure 4 shows, we use regular expressions to capture external representations. The *Price* data frame, for example, captures instances of this concept such as "$4500" and "17,900". A data frame's context keywords are also regular expressions. The *Price* data frame in Figure 4, for example, includes context keywords such as "asking" and "negotiable". In the context of one of these keywords in a car ad, if a

```
Price
   internal representation: Integer
   external representation: \$?(\d+ | \d?\d?\d,\d\d\d)
   context keywords: price | asking | obo | neg(\.|otiable) | ...
   ...
   LessThan(p1: Price, p2: Price) returns (Boolean)
   context keywords: less than | < | or less | fewer | ...
   ...
end

Make
   external representation: CarMake.lexicon
   ...
end
```

Fig. 4. Sample data frames for car ads ontology

number appears, it is likely that this number is a price. The operations of a data frame can manipulate a concept's instances. For example, the *Price* data frame includes the operation *LessThan* that takes two instances of *Price* and returns a *Boolean.* The context keywords of an operation indicate an operation's applicability; context keywords such as "less than" and "<", for example, apply to the *LessThan* operation. Sometimes external representations are best described by lexicons or other reference sets. These lexicons or reference sets are also regular expressions, often simple lists of possible external representations, and can be used in place of or in combination with regular expressions. In Figure 4, *CarMake.lexicon* is a lexicon of car makes, which would include, for example, "Toyota", "Honda", and "Nissan" and potentially also misspellings (e.g. "Volkswagon") and abbreviations (e.g. "Chev" and "Chevy").

We can apply an extraction ontology to obtain a structured representation of the unstructured information in a relevant document. For example, given the car-ads extraction ontology and one of the Nissan ads in Figure 1:

> **'93 NISSAN** Model XE, $900, Air Conditioning, new tires, sweet cherry red.
> For listings call 1-800-749-8104 ext. V896.

we can extract "**'93**" as the *Year*, "**NISSAN**" as the *Make*, "XE" as the *Model*, "$900" as the *Price*, "red" as the *Color*, both "Air Conditioning" and "new tires" as *Features* with "Air Conditioning" also being an *Accessory*, and "1-800-749-8104" as the *PhoneNr.* As part of the extraction, the conversion routines in the data frames convert these extracted values to canonical internal representations, so that, for example, "**'93**" becomes the integer *1993* and "$900" becomes the integer *900.*

2.2 Annotation Through Instance Recognition Semantics

Information-extraction ontologies are well positioned to satisfy the requirements of semantic annotation. Not only do they provide the intentional-level semantics found in typical ontologies, but they also provide the instance recognition semantics needed to connect individual data items found in ordinary web pages with the typical intentional-level semantics.

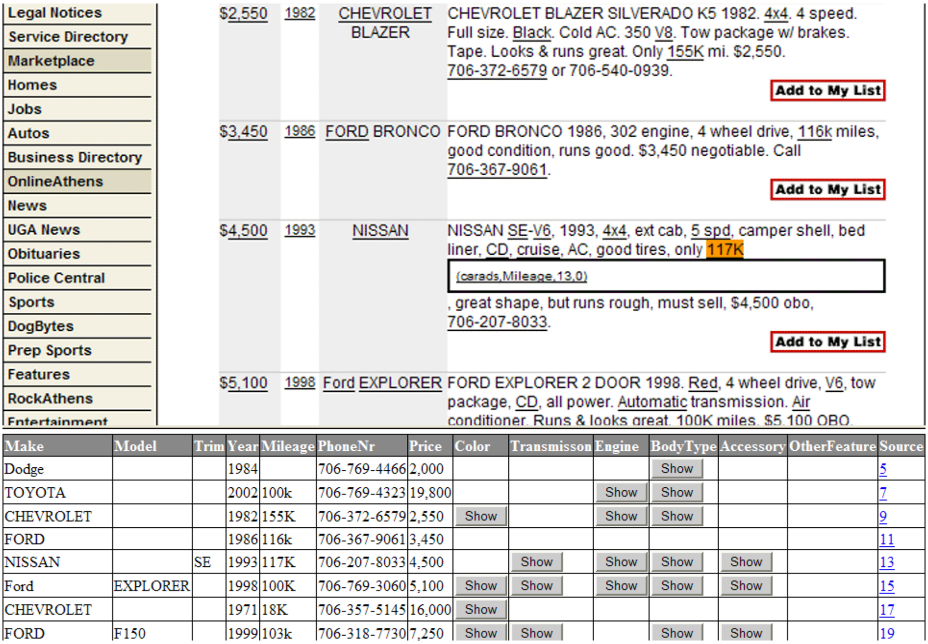
Figure 4 exemplifies the fundamental idea. The external representations describe textual instantiation patterns of a concept. Added to these instantiation patterns, we provide regular expressions for context and keyword phrases, which aid in correctly classifying instantiation patterns that may be similar in several different data frames.

This approach stands in stark contrast to a typical automated semantic annotation paradigm (e.g., the approaches in [1], [4], [8], [9], [12], and [16]), which do not use extraction ontologies. Although results are encouraging for these automated semantic annotators, there are problems in these annotation paradigms. A complete annotation process using typical non-ontology-based IE tools contains three basic procedures: extraction, alignment, and annotation. Although researchers have neither fully resolved the issues with the first procedure nor decided on the best solution for the third procedure, it is the second procedure that has become the most critical for those attempting to adapt IE tools to annotate current web pages for the semantic web. It is nontrivial to align extraction categories in an IE wrapper with concepts defined in semantic-web ontologies. Sheth and Ramakrishnan believe this "concept disambiguation" problem is a major issue for the semantic annotation [14]. Furthermore, Kiryakov et al. think that this requirement of post-extraction alignment is the "main drawback" of current automated annotation approaches [9]. They suggest that we need to integrate domain ontologies with extraction engines to solve the problem and propose this as a direction for future work [9]. Indeed, this is the approach we take. Since information-extraction ontologies represent extraction categories with ontologies, we can combine the problems of data recognition and concept disambiguation and thus simplify the structure of the semantic annotation problem.

3 IE-Based Semantic Web Annotation

Generally speaking, there are two ways to represent annotated data instances: *explicit annotation*, which adds special tags that bind tagged instances in a web page to an externally specified ontology, and *implicit annotation*, which adds nothing explicit to the document, but instead extracts instance position information as well as the data instances and stores them in an externally specified knowledge base. In our prototype, we have implemented both explicit and implicit annotation.

Using explicit annotation, we have created an online demonstration [3] of our semantic annotation tool. Figure 5 is a screen shot showing that our system has extracted specific information from a web site containing car ads and has, in addition, annotated the web page so that we can highlight extracted information with the hover feature of CSS. The hover feature is only for the demonstration. For the annotation itself, we include a four-tuple in each tag for every recognized data instance x. This four-tuple uniquely identifies (1) the ontology used for annotation (in case there are several for the same document), (2) the concept within the



$2,550	1982	CHEVROLET BLAZER	CHEVROLET BLAZER SILVERADO K5 1982. 4x4. 4 speed. Full size. Black. Cold AC. 350 V8. Tow package w/ brakes. Tape. Looks & runs great. Only 155K mi. $2,550. 706-372-6579 or 706-540-0939.
$3,450	1986	FORD BRONCO	FORD BRONCO 1986, 302 engine, 4 wheel drive, 116k miles, good condition, runs good. $3,450 negotiable. Call 706-367-9061.
$4,500	1993	NISSAN	NISSAN SE-V6, 1993, 4x4, ext cab, 5 spd, camper shell, bed liner, CD, cruise, AC, good tires, only 117K (carads,Mileage,13,0) , great shape, but runs rough, must sell, $4,500 obo, 706-207-8033.
$5,100	1998	Ford EXPLORER	FORD EXPLORER 2 DOOR 1998. Red, 4 wheel drive, V6, tow package, CD, all power. Automatic transmission. Air conditioner. Runs & looks great. 100K miles. $5,100 OBO.

Make	Model	Trim	Year	Mileage	PhoneNr	Price	Color	Transmisson	Engine	BodyType	Accessory	OtherFeature	Source
Dodge			1984		706-769-4466	2,000				Show			5
TOYOTA			2002	100k	706-769-4323	19,800			Show	Show			7
CHEVROLET			1982	155K	706-372-6579	2,550	Show		Show	Show			9
FORD			1986	116k	706-367-9061	3,450							11
NISSAN		SE	1993	117K	706-207-8033	4,500		Show	Show	Show	Show		13
Ford	EXPLORER		1998	100K	706-769-3060	5,100	Show	Show	Show	Show	Show		15
CHEVROLET			1971	18K	706-357-5145	16,000	Show						17
FORD	F150		1999	103k	706-318-7730	7,250	Show	Show		Show	Show		19

Fig. 5. Page Annotation Demo: Car Ads from Athens Site (Hovering on 117K)

ontology to which x belongs, (3) the record number for x so that the system knows which values relate together to form a record, and (4) a value number within the record in case more than one instance of the concept can appear within a record, as happens in our ontology for car ads, for example, with *Feature*, which can have multiple values in a single record. Thus, for example, we annotate the value *117K* in Figure 5 by <span class="(CarAds,Mileage,13,0)">117K</span>. Here *CarAds* is the ontology, *Mileage* is the concept, *13* is the record number, and *0* is the value number. *Span* annotations along with a URL specifying an OWL ontology [13] allow the system to create the equivalent of a populated semantic ontology for each annotated page.

For implicit annotation, we start by generating an OWL ontology from an extraction ontology. Then we create an RDF data file to store annotated data instances based on the domain declaration defined in the OWL ontology. Figure 6 shows a portion of an implicit annotation for the Athens web page in Figure 1. When we do implicit annotation, we also cache a copy of the web page so that we can guarantee that the instance position information is correct. In Figure 6, we first declare several namespaces of referenced ontologies and web pages. Specifically, we include an *ontos* namespace, which provides general system information, a namespace referencing the ontology we use for annotation (here *carad*), and a *webpage* namespace for the annotated web pages. For each car, we store its canonical data values with their respective attribute names. For a lexical concept, such as mileage in Figure 6, we store its original value in the source text (*117K*), its

```
<rdf:RDF ... xmlns:ontos=“http://www.deg.byu.edu/ontology/ontosBasic#”
             xmlns:carad=“http://www.deg.byu.edu/ontology/carad#”
             xmlns:webpage=“http://www.deg.byu.edu/demos/...” ... >
...
  <rdf:Description rdf:about=“&webpage;CarIns13”>
    <carad:Mileage>117000</carad:Mileage>
    <carad:Price>4500</carad:Price>
    <carad:Make>Nissan</carad:Make>
    <carad:Year>1993</carad:Year>
    ...
  </rdf:Description>
  <rdf:Description rdf:about=“&webpage;Mileage13”>
    <ontos:ValueInText>117K</ontos:ValueInText>
    <ontos:CanonicalValue>117000</ontos:CanonicalValue>
    <ontos:CanonicalDataType>xsd:integer</ontos:CanonicalDataaype>
    <ontos:CanonicalDisplayValue>117,000</ontos:CanonicalDisplayValue>
    <ontos:Offset> 37733 </ontos:Offset>
  </rdf:Description>
...
</rdf:RDF>
```

Fig. 6. Implicit Annotation for Car Ads Web Page

canonical internal value (*117000*) and type (*integer*), and its canonical display value (*117,000*). We use canonical internal values (together with type information) in SPARQL queries and use canonical display values when returning results to users (as in Figure 2). We also store offset values in the cached web page (e.g. *37733* is the actual offset of the extracted instance “117K”). The RDF document in Figure 6 fully annotates the Athens web page in Figure 1.

4 Querying Annotated Semantic Web Pages

Given an implicit annotation in an RDF file, we can query the file and thus query the annotated web page. Because we store information in an RDF file, we can use SPARQL to query the information directly, as we explain in Section 4.1. Ordinary users, however, will not be able to write queries in SPARQL. We therefore argue in Section 4.2 that a more user-friendly mechanism is needed and further show that information-extraction ontologies may give us a reasonable way to provide the needed user-friendly mechanism.

4.1 SPARQL for Implicitly Annotated Semantic Web Pages

Figure 7 shows a SPARQL rendition of our sample query, “Find me a red Nissan for under $5000 – it should be a 1990 or newer and have less than 120K miles on it.” The query is written over the RDF file in Figure 6 that annotates the web page. The *PREFIX* clause associates a prefix label with an IRI (a generalization of URIs that is fully compatible with URIs and URLs). The prefix label becomes a local namespace abbreviation for the address specified by the IRI. The *SELECT* clause names the result

```
PREFIX  carad: <http://www.deg.byu.edu/ontology/carad#>
PREFIX  xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT  ?make ?color ?price ?year ?mileage
WHERE   { ?x carad:Make ?make . FILTER (?make = "Nissan") .
    OPTIONAL {?x carad:Color ?color} . FILTER (?color = "red") .
    OPTIONAL {?x carad:Price ?price} . FILTER (xsd:integer(?price) < 5000) .
    OPTIONAL {?x carad:Year ?year} . FILTER (xsd:integer(?year) >= 1990) .
    OPTIONAL {?x carad:Mileage ?mileage} .
                                FILTER (xsd:integer(?mileage) < 120000) }
```

Fig. 7. SPARQL Query to Search an Annotated Web Page

variables. The first clause in the *WHERE* clause requires the car bound to x to have *Make* equal to *NISSAN*. Each *OPTIONAL* clause checks whether a corresponding extracted value satisfies certain constraints. The keyword *OPTIONAL* allows the content to be unbound. Otherwise, however, any bound value must satisfy the constraints in the corresponding *FILTER* clause. To perform semantic-web searches, we apply this query to all documents that are applicable to the given domain, collect the results, and display them to the user in a tabular format as Figure 2 shows.

The reason we make our conditions *OPTIONAL* is that optional elements might not be present in some of the records. Thus, as is the case with the ordinary web, our semantic-web queries may return irrelevant results. For example, suppose a car ad does not list the car's color, but otherwise satisfies the user's constraints. Rather than miss a potential object of interest, we allow optional elements to be missing, and we return the partial record with the query results. It would not be hard to allow users to override this behavior and require the presence of all concepts in each of the query results.

4.2 IE-Based Semantic Web Queries

For researchers and developers, SPARQL is a fine choice as a query language for the semantic web. On the other hand, few end users will have the ability, patience, or interest to learn to write SPARQL queries. A practical semantic-web query solution must be sufficiently expressive while also being easy to use. We believe that, like current web search engines, semantic-web searches will migrate to free-form text. Because it is impossible to execute free-form queries directly, mapping from free-form queries to executable queries is necessary.

Our approach can be characterized as an *information-extraction, ontology-based, natural-language-like approach.* The essence of the idea is to (1) extract constants, keywords, and keyword phrases in a free-form query; (2) find the ontology that matches best; and (3) embed the query in the ontology yielding (3a) a join over the relationship-set paths connecting identified concepts, (3b) a selection on identified constants modified by identified operators, and (3c) a projection on mentioned concepts.[3]

[3] See [17] for a full explanation. The theoretical underpinnings of this approach are found in the "window functions" explained in [11].

Consider our running example, where the user specifies, "Find me a red Nissan for under \$5000 – it should be a 1990 or newer and have less than 120K miles on it." The extraction ontology from our library that best matches this query is the car-ads ontology. When we apply our car-ads extraction ontology to this sentence, we discover that the desired object has restrictions on five concepts: color, make, price, year, and mileage. For string-valued concepts (color and make), we can test equality (either equal or not equal). Since there are no keyword phrases in the query that indicate negation, we search for objects where *Color=red* and *Make=Nissan*. For numeric concepts (price, year, and mileage), we can test ranges. Associated with each operator in a data frame are keywords or phrases that indicate when the operator applies. In this case, "under" indicates $<$ (a less-than comparison), "or newer" indicates $\geq$, and "less than" indicates $<$. So in our example, we must search for $Price < 5000$, $Year \geq 1990$, and $Mileage < 120000$. Recall, from our discussion in Section 2, that our data frames specify operators that convert a string to a canonical internal representation and to a canonical representation for display. Thus, for example, *"120K"* becomes the integer *120000* as its canonical internal representation and the string "120,000" as its canonical display value. We therefore are able to apply standard conditions and *FILTER* clauses to compose a SPARQL query. Because web data is stored using an open world assumption, we should not reject an answer when a data value is not present. Hence, by default we add *OPTIONAL* before each generated condition. There is, however, another factor that decides the *OPTIONAL* before a generated condition. When a minimum participation constraint in the extraction ontology is "1", the corresponding generated condition becomes mandatory instead of optional. For example, in Figure 3, each *Car* must have one and only one *Make*. We therefore remove the default *OPTIONAL* from the generated condition of *Make*. Figure 8 shows the particular concept conditions for our example. Given a set of concept conditions, we can readily generate, rather than manually write, the SPARQL query in Figure 7.

5 Evaluation

We provide two types of evaluation—an objective evaluation of annotation accuracy and a subjective evaluation giving our view of what it would take to make our prototype system viable and practical.

5.1 Accuracy Evaluation

We are interested, of course, in how accurately an annotation system binds real-world data to the concepts defined in annotation ontologies. Since our annotation results depend, and only depend, on our ability to correctly extract information, we can apply the traditional information extraction (IE) evaluation metrics, precision and recall, to evaluate performance accuracy. We point out, however, that this is not the case for a traditional non-ontology-based IE process. For non-ontology-based IE annotators, calculations of precision and recall are according to either self-defined or machine-learned extraction categories. But for semantic annotation, we need to compute precision and recall with respect to the concepts defined in a domain ontology. Therefore, for systems that use a non-ontology-based IE engine, there are two precision and recall metrics. One evaluates the performance of the IE process itself, and

Name	Operator	Value	Optional
Color	$=$	*red*	*true*
Make	$=$	*Nissan*	*false*
Price	$<$	*5000*	*true*
Year	$\geq$	*1990*	*true*
Mileage	$<$	*120000*	*true*

Fig. 8. Filters Extracted from Natural-Language User Query

the other evaluates how well the system maps these extraction categories to the concepts defined in an ontology. The final precision and recall values are the products of the two respective precision and recall values. This is not required for annotation systems that use ontology-based IE engines, such as ours. Because of the integration of ontologies into the extraction process itself, the evaluation of precision and recall for the semantic annotation system is the same as the evaluation of precision and recall for the original ontology-based IE tool.

Although the study of semantic annotation is still a new research topic, researchers have studied information extraction for more than a decade, and so have we. Over the course of many years, we have developed our ontology-based IE tool and have tested it on various domains, each with dozens of real-world web pages. Among them, there are some simple, unified domains like automobile sales and apartment rentals, and there are complicated or loosely unified domains like genealogy and obituaries.

Based on approximately 20 domains with which we have experimented we summarize our experience as follows. In simple, unified domains we typically achieve close to 100% precision and recall in almost all fields, while in more complicated or loosely unified domains, the precision and recall for some fields falls off dramatically. For obituaries, for example, we were only able to achieve about 74% precision for relatives of the deceased and only about 82% recall for recognizing funeral addresses. In general, within nearly 20 domains that contain in total over 200 different object sets, our extraction engine typically achieves at least 80% accuracy for both precision and recall values on most fields. For over half of the domains, the precision and recall values were above 90%.

We have recently been able to obtain some initial results for IE-based query conversion [17]. Four subjects each provided five queries on five domains (car ads, real estate, countries, movies, and diamonds) for a total of 100 queries. The recall for identifying concept values to be returned was 89% and for correctly generating conditions was 75% while the corresponding precision values were respectively 89% and 88%. Overall, the system interpreted 47% of the queries with perfect accuracy while interpreting an additional 49% with partial accuracy.

5.2 Practical Considerations

Beyond accuracy, there are several criteria that a practical semantic annotation system should satisfy, such as generality, resiliency, and conformance to standards. In contrast with precision and recall measures, it is harder to establish objective metrics for these practical considerations. We cannot, however, ignore these important criteria, since the success of a semantic annotation system depends on them.

Our first practical consideration is generality of the semantic annotation approach. In other words, what is the range of pages for which the annotation system is effective? Because we use an ontology-based IE engine, our prototype system targets data-rich web pages that each have a relatively narrow domain [6]. There is no particular restriction that limits applicability, but as the domain of a page broadens, our approach becomes less accurate because the instance recognition semantics overlap more and become harder to segment. This issue is not unique with our approach (see, for example, [12]). Fortunately, narrow-domain, data-rich pages are quite common on the web (consider shopping, news, and product portals, for example).

Within an application domain, our semantic annotation approach works best on semi-structured web pages containing multiple records that are laid out in a straightforward way. A multi-record collection lets our system cross-validate the correctness of recognized data instances. The approach, however, also works on single-record web pages and complex web pages with complicated table structures. Although our method is also applicable to fully unstructured natural-language text, our experiments show that performance is usually lower for these types of pages. Unlike other semantic annotators (such as [8] and [9]), there are no typical natural-language-processing (NLP) techniques encoded in our ontology-based data-recognition program. A question we expect to explore in the future is whether a hybrid system that also uses NLP techniques will increase the generality of our approach.

Another practical consideration is the resiliency of an annotation system. Web pages change often, both in terms of current content and physical layout. If such changes break the underlying automatic annotator, someone will have to work to maintain the annotation system, and such an approach will ultimately fail to scale. Our approach is resilient to web page layout changes, and thus we minimize the need for wrapper maintenance in the information-extraction layer of the system [10]. A trade-off for resiliency is that our current system sacrifices some execution speed (and possibly even some accuracy). To address this problem, we have proposed—and are working to develop—a two-layer semantic annotation architecture that will divide the work more efficiently into an upper-layer set of structural annotators and base-layer conceptual annotators. Each layer will be optimized for its particular task.

Another practical consideration is adherence to accepted standards. The reason we annotate pages in the semantic web is so we can use them. Any system that does not conform to semantic web standards will not be interoperable, and thus will not be used. Thus, we convert our proprietary OSMX ontologies to standard OWL ontologies [13] when we generate annotations. Most recent semantic annotation approaches adopt a similar strategy. Researchers using implicit annotation (where annotations are stored separately from source pages) typically use either RDF [9] or DAML+OIL [8].

6 Concluding Remarks

We have presented an approach to semantic web-page annotation that is based on the use of data-extraction ontologies. We have argued that ontology-based information-extraction engines can provide a solid foundation for an automated semantic-web annotation tool. Ontology-based IE engines provide two fundamental advantages: (1) they include declared instance recognition semantics, and (2) they extract information directly into an annotation ontology. In our experiments, both precision and recall are running at roughly 85% to 90% for each of the individual lexical concepts in an extraction ontology. Our prototype implementation supports both internal and external annotation. We can directly query our external annotation with SPARQL. We can also generate SPARQL queries from free-form text input, and we therefore provide a way for ordinary users to query annotated semantic-web pages. In initial experiments with the query generator, 47% of the queries submitted by subjects returned fully correct results, and all but 4% returned some useful results. We are currently working to improve the quality of these generated queries.

The future of the semantic web is bright, but delivering on its vision will not be easy. Effective deployment of the semantic web requires some way to automatically accommodate the huge quantity of existing data-rich web pages on the ordinary web, and some way to handle ordinary user requests. Our approach addresses these challenges.

References

1. L. Arlotta, V. Crescenzi, G. Mecca, and P. Merialdo, “Automatic annotation of data extracted from large web sites,” *Proc. Sixth International Workshop on the Web and Databases (WebDB 2003)*, pp. 7-12, San Diego, California, June 2003.
2. T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” *Scientific American*, vol. 36, no. 25, pp. 34-43, May 2001.
3. Homepage, *BYU Data Extraction Group*, URL: http://www.deg.byu.edu.
4. S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, K.S. McCurley, S. Rajagopalan, A. Tomkins, J.A. Tomlin, and J.Y. Zien, “A Case for Automated Large Scale Semantic Annotations,” *Journal of Web Semantics*, vol. 1, no. 1, pp. 115–132, December 2003.
5. D.W. Embley and B.D. Kurtz and S.N. Woodfield, *Object-oriented Systems Analysis: A Model-Driven Approach*, Prentice Hall, Englewood Cliffs, New Jersey, 1992.
6. D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith, “Conceptual-model-based data extraction from multiple-record web pages,” *Data & Knowledge Engineering*, vol. 31, no. 3, pp. 227-251, November 1999.
7. T.R. Gruber, “A translation approach to portable ontology specifications,” *Knowledge Acquisition*, vol. 5, no. 2, pp. 199-220, 1993.
8. S. Handschuh, S. Staab, and F. Ciravegna, “S-CREAM Semi-automatic CREAtion of Metadata,” *Proc. European Conference on Knowledge Acquisition and Management (EKAW-2002)*, pp. 358–372, Madrid, Spain, October, 2002.
9. A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff, “Semantic Annotation, Indexing, and Retrieval,” *Journal of Web Semantics*, vol. 2, no. 1, pp. 49–79, December 2004.
10. K. Lerman, S. N. Minton, and C. A. Knoblock, “Wrapper maintenance: A machine learning approach,” *Journal of Artificial Intelligence Research*, vol. 18, pp.149–181, 2003.
11. D. Maier, *The Theory of Relational Databases*, Computer Science Press, Inc., Rockville, Maryland, 1983.

12. S. Mukherjee, G. Yang, and I.V. Ramakrishnan, "Automatic Annotation of Content-Rich HTML Documents: Structural and Semantic Analysis," *Proc. Second International Semantic Web Conference (ISWC 2003)*, pp. 533–549, Sanibel Island, Florida, October, 2003.
13. W3C (World Wide Web Consortium) *OWL Web Ontology Language Reference*, http://www.w3.org/TR/owl-ref/.
14. A. Sheth and C. Ramakrishnan, "Semantic (Web) technology in action: Ontology driven information systems for search, integration and analysis," *IEEE Data Engineering Bulletin*, vol. 26, no. 4, pp. 40-48, December 2003.
15. W3C (World Wide Web Consortium), *SPARQL Query Language for RDF*, February 2006. URL: http://www.w3.org/TR/rdf-sparql-query/.
16. M. Vargas-Vera, E. Motta, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna, "MnM: Ontology Driven Tool for Semantic Markup," *Proc. Workshop Semantic Authoring, Annotation & Knowledge Markup (SAAKM 2002)*, pp. 43–47, Lyon, France, July, 2002.
17. M. Vickers, "Ontology-Based Free-Form Query Processing for the Semantic Web," Masters Thesis, Brigham Young University, Provo, Utah, June 2006.

HStar - A Semantic Repository for Large Scale OWL Documents

Yan Chen, Jianbo Ou, Yu Jiang, and Xiaofeng Meng

{chenyan8, oujianbo, jiangyu, xfmeng}@ruc.edu.cn
School of Information, Renmin Univ. of China, China, 100872

Abstract. HStar is implemented to support large scale OWL documents management. Physical storage model is designed on file system based on semantic model of OWL data. Inference and query are implemented on such physical storage model. Now HStar supports characters of OWL Lite and we try to adopt strategy of partial materializing inference data, which is different from most of existing semantic repository systems. In this paper we first give the data model which HStar supports, then give an analysis of our inference strategy; storage model and query process are discussed in detail; experiments for comparing HStar and related systems are given at last.

1 Introduction

RDF(S) standard is firstly proposed by W3C to support research and application of semantic web. It can be used to describe Ontology and metadata with very limited express ability. To support more complicated application of semantic web, OWL standard, which is built on RDF(S), is brought forward. OWL imports more vocabu-laries and rules and is divided into three sub languages: OWL-Lite, OWL-DL and OWL-Full based on the express ability. Semantic web needs high performance se-mantic repository for OWL documents. Now there are many prototype systems, most of which depend on relation database and are designed for RDF(S) documents. From RDF(S) to OWL, more semantic rules make performance of these systems depraved dramatically. This can be proved by our experiment of Sesame [1] using database storage model. Relation database has a single storage model, which cannot satisfy complex data model of OWL data, e.g. the hierarchy relation in OWL data cannot be represented by relation table directly. Relation database can only use logical pointer, not physical pointer, to link different entity in OWL data. Most systems completely materialize inference data to reduce join operation of logical pointers. For such method, more complicated storage strategy is needed to support update operation and this will affect system performance seriously, e.g. Sesame [1] constructs large de-pendent relations among entities of OWL data after loading operation and this is a waste of time. Completely materializing inference data is not fit for large scale OWL documents, because large redundancy data will be produced and this will also affect system performance, especially for loading operation. This has been proved by our experiment discussed in section 6. HStar designs physical storage model, which is independent of relation database and based on characters of OWL Lite data. Most of inference is processed during query processing

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 415–428, 2006.
© Springer-Verlag Berlin Heidelberg 2006

time to avoid storing large scale inference data. Our aim is to improve performance of semantic repository and provide the possibility of managing large scale semantic data.

2 Relate Work

Along with more and more popular research of semantic web, many semantic repositories have been developed. All of them can be divided into three categories based on the persistent strategy they use: RDB (Relational Database)-based, File system-based and Memory-based. Because RDB has been fully studied these years, RDB-based systems are in the majority, like Sesame [1], DLDB-OWL [6], RStar [5] and so on. Sesame provides a general storage interface and implements storage method on MySQL, Oracle and so on. File system-based and memory-based storage methods have also been implemented. Sesame provides two logical storage models: RDF schema and RDFS schema. No inference is supported for RDF schema. For RDFS schema, user can use default inference function defined by Sesame, but this is limited to inference rules defined in RDFS. Moreover, user can also use self-defined inference rules, which makes Sesame have good extensibility. From RDF(S) to OWL, only the self-defined inference rules change. But from experiment, we can observe that large number of rules is needed to express complete OWL semantics and when loading data, performance is very bad for doing complete inference based on such rules. Therefore, it can't be used to manage large scale OWL data. DLDB-OWL uses MS Access as its persistent platform and uses inference engine FaCT. It declares high performance for large scale OWL data, but has limited inference ability. From ex-periment we can observe that DLDB-OWL cannot get any answers for some queries. OWLim [3] is a typical memory-based system. It supports more semantic rules than any other systems. OWLim uses Sesame's general storage interface and it has higher performance than Sesame's own memory-based storage module. To support persis-tent storage of semantic data, OWLim uses a simple file format, named "N-triples" and provides backup function. But when do query and inference processing, all data will be read from hard disk into memory. From experiment, we can observe that OWLim cannot handle OWL documents which size is larger than 100MB on general computer hardware. Because OWLim supports most of the semantic rules in OWL Lite, we use it as benchmark of query completeness in our experiment. To support large scale semantic data management, HStar is built on file system and do query and inference processing on physical storage model. Very small part of inference data is materialized and almost the same semantic rules are supported as OWLim. Only one query has less answers than OWLim when do test queries of Lehigh University Benchmark.

3 Data Model

To give better description of HStar's functions, we formalized data model of OWL supported by HStar. This data model has summarized most of the characters of OWL Lite. Our storage, inference and query processing strategies are all based on the data model.

D: all data in OWL document as format<subject property object>

$L = \{C, P, I, R_C, R_P, R_{CP}, R_I, R_{CI}, T_P\};$

$C = \{URI_i | \exists < URI_i \text{ rdf:type owl:Class } > \in D\};$

$P = \{URI_i | \exists < URI_i \text{ rdf:type owl:ObjectProperty } > \vee$
$\quad \exists < URI_i \text{ rdf:type owl:DatatypeProperty } > \in D\};$

$I = \{URI_i | URI_i \notin C \text{ and } URI_i \notin P\};$

$R_C = \{C_i \prec C_j | C_i, C_j \in C \wedge \exists < C_j \text{ rdfs:subClassOf } C_i > \in D\} \cup$
$\quad \{C_i \equiv C_j | \exists < C_i \text{ owl:equivalentClass } C_j > \in D\};$

$R_P = \{P_i \prec P_j | P_i, P_j \in P \wedge < P_j \text{ rdfs:subPropertyOf } P_i > \in D\} \cup$
$\quad \{P_i \equiv P_j | \exists < P_i \text{ owl:equivalentProperty } P_j > \in D\} \cup$
$\quad \{P_i \leftrightarrow P_j | \exists < P_i \text{ owl:inverseOf } P_j > \in D\};$

$R_{CP} = \{[P_i, C_j] | \exists < P_i \text{ rdfs:domain } C_j > \in D \vee \exists < P_i \text{ rdfs:range } C_j > \in D\};$

$R_I = \{[URI_i, URI_j] | \exists < URI_i \; P_x \; URI_j >, P_x \in P \in D\} \cup$
$\quad \{URI_i \equiv URI_j | \exists < URI_i \text{ owl:sameAs } URI_j > \in D\};$

$R_{CI} = \{[URI_i, C_j] | \exists < URI_i \text{ rdf:type } C_j > \in D\};$

$T_P = P_T \cup P_S \cup P_F \cup P_{IF};$

$P_T = \{P_i | P_i \in P \wedge < P_i \text{ rdf:type owl:TransitiveProperty } > \in D\};$

$P_S = \{P_i | P_i \in P \wedge < P_i \text{ rdf:type owl:SymmetricProperty } > \in D\};$

$P_F = \{P_i | P_i \in P \wedge < P_i \text{ rdf:type owl:FunctionalProperty } > \in D\};$

$P_{IF} = \{P_i | P_i \in P \wedge < P_i \text{ rdf:type owl:InverseFunctionalProperty } > \in D\};$

OWL data has been divided into three categories by this data model: one consists of C, P, I, which respectively represent OWL Class, OWL Property and Individual Resource; one consists of R_C, R_P, R_I, R_{CP}, R_{CI}, which respectively represent relation of elements in C, relation of elements in P, relation of elements in I, relation between elements in C and elements in P, relation between elements in C and elements in I; the last one is T_P, which represents characters defined on OWL Property, including transitive P_T, symmetric P_S, functional P_F and inverse functional P_{IF}. C, P, R_C, R_P, R_{CP} and T_P are used to define Ontology and they are always stable. Most of OWL data focus on R_I, which use Ontology to describe type and relation information of elements in I. Completeness of inference includes two aspects: one is to get complete relation of R_C and R_P, the other is to get complete relation of R_I and R_{CI}. The former represents complete ontology and the latter represents complete ontology instances.

4 Analysis of Inference Completeness

We mentioned above that inference completeness is to get complete R_C, R_P, R_I and R_{CI}. Below we give a detailed discussion for them respectively:

4.1 Completeness Analysis of R_C, R_P

Elements in R_C have two relations: $C_i \prec C_j$ represents inheritance; $C_i \equiv C_j$ represents equivalence. Inheritance is transitive. Equivalence is transitive and symmetric. Because equivalence affects inheritance, complete equivalence relation should be computed first. Complete R_C should satisfy:

(a) $\forall C_i \in C$, can get all $\{C_j | C_j \in C \wedge \exists$ (explicit or implicit) $C_i \equiv C_j\}$;

(b) $\forall C_i \in C$, can get all $\{C_j | C_j \in C \wedge \exists$ (explicit or implicit) $C_j \prec C_i\}$;

(c) $\forall C_i \in C$, can get all $\{C_j | C_j \in C \wedge \exists$ (explicit or implicit) $C_i \prec C_j\}$;

There are three methods to guarantee requirement (a): The first is to store all explicit $C_i \equiv C_j$ and get all relevant $\{C_i \equiv C_j\}$ to construct equivalent set when query; The second is to store all explicit and implicit $C_i \equiv C_j$. There will be no implicit data left and equivalent set does not need to be built. The third is to store equivalent set directly on hard disk. The second method uses redundancy data to improve search performance but adds maintenance cost. The third method not only avoids redundancy data, but also can get equivalent set directly. It is suitable for managing large equivalence relation. In general, equivalence relation in OWL is quite few. So HStar adopts the first method.

For inheritance relation $C_i \prec C_j$, transitive character makes $C_i \prec C_j \prec C_k \Rightarrow C_i \prec C_k$. Requirement (b) and (c) are all related with it. There are also two methods for these two requirements: One is for every transitive chain, compute all implicit inheritance relation and put them into storage system. E.g. if use $C_i \leftarrow C_j$ represents $C_i \prec C_j$ and suppose there are inheritance relations in fig.1.

There are four transitive chains in fig.1: $C_i \prec C_j \prec C_l, C_i \prec C_j \prec C_m, C_i \prec C_k \prec C_m, C_i \prec C_k \prec C_n$. We can compute three implicit inheritance relations from these chains: $C_i \prec C_l, C_i \prec C_m, C_i \prec C_n$. For large inheritance relation, such method will produce too much redundancy data. Computation complexity is $O(n^2)$ (n is the number of elements in inheritance relations) and it will be a hard work to maintain the redundancy data. The other method is using tree storage structure to represent inheritance relations.

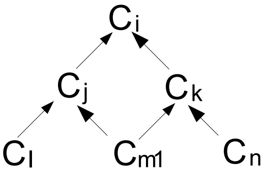


Fig. 1. An example of inheritance relation

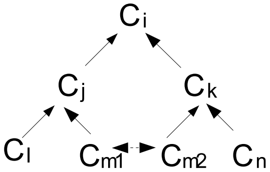


Fig. 2. Tree structure for Fig.1

Node C_m in fig.1 splits into nodes C_{m1} and C_{m2}. Node C_{m1} copies all information of Cm and node C_{m2} is a reference of C_{m1}. If it is required to find all C_x, which satisfy $C_x \prec C_m$, first locate node C_{m1} in Fig.2, get all ancestors of C_{m1}, i.e. $\{C_i, C_j\}$, and then get all ancestors of C_{m2}, i.e. $\{C_i, C_k\}$, union the two result sets and remove the duplicate, we can get $\{C_i, C_j, C_k\}$. Such method avoids computing redundancy data, but needs native tree storage on hard disk. HStar adopts this method.

Inheritance relation and equivalence relation defined in R_P are same as those in R_C in essence. HStar uses same method to deal with them. Besides these, there is another relation defined, i.e. $\{P_i \leftrightarrow P_j | < P_i$ owl:inverseOf $P_j > \in D\}$. This relation will only bring implicit data in R_l according to OWL semantic definition. So we will discuss it later.

4.2 Completeness Analysis of R_I

From definition of R_I, we can see there are two sub-relations in it. We give their definitions below:

$$R_{I1} = \{[URI_i, URI_j] | \exists < URI_i \, P_x \, URI_j > \in D\}$$
$$R_{I2} = \{URI_i \equiv URI_j | \exists < URI i \, \text{owl:sameAs} \, URI_j > \in D\}$$

R_{I2} defines equivalence relation which affects completeness of R_{I1}, just as equivalence relation in R_C does. Besides that user can directly define R_{I2}, property that is element of P_F or P_{IF} can also infer R_{I2} relation. The inference rules are defined below:

$$< URI_i P_x URI_k > \wedge < URI_j P_x URI_k > \wedge P_x \in P_{IF} \Rightarrow URI_i \equiv URI_j$$
$$< URI_k P_x URI_i > \wedge < URI_k P_x URI_j > \wedge P_x \in P_F \Rightarrow URI_i \equiv URI_j$$

So complete R_{I2} needs to apply rules above to every element in P_F and P_{IF}. And the process needs to do iteratively. E.g. suppose $P_x \in P_F$, a, b, c, d respectively represent an URI and $\exists\{< a \, P_x \, b >, < a \, P_x \, c >, < b \, P_x \, d >, < c \, P_x \, a >\} \in D$. According to rules above, we can get $< a \, P_x \, b > \wedge < a \, P_x \, c > \Rightarrow b \equiv c$. But the process cannot terminate now, because $b \equiv c$ also affects existed data. With this consideration, we can get $< b \, P_x \, d > \wedge < c \, P_x \, a > \Rightarrow d \equiv a$. The process needs to do iteratively until no new equivalence relations are generated. Storage method of R_{I2} is the same as equivalence relation of R_C.

Completeness of R_{I1} is mainly determined by characteristic of P_x. If $P_x \in P_T$ or $P_x \in P_S$ or P_x has inheritance or equivalence relation in R_p, it will bring implicit data into R_{I1}. If there exist P_x satisfying $P_x \in P_T \wedge P_x \in P_S$, we treat such P_x as an equivalent relation.

There is a condition that is not defined definitely in OWL semantic. If $\{P_x \prec P_y \in R_p$ or $P_y \prec P_x \in R_p\}$ and $\{P_x \in P_T$ or $P_x \in P_S$ or $P_x \in P_F$ or $P_x \in P_{IF}\}$, whether $P_y \in P_T$ or $P_y \in P_S$ or $P_y \in P_F$ or $P_y \in P_{IF}$ is not defined. So HStar does not consider the interaction effect between R_p and T_p.

Under the precondition above, completeness of R_{I1} can be considered from P_T, P_S and R_P respectively:

1. P_T defines transitive character which is equivalent to inheritance relation of R_C in essence. HStar adopts the same method to deal with P_T.
2. PS defines symmetric relation and related rule in OWL is
   $P_x \in P_S \wedge < URI_i \, P_x \, URI_j > \Rightarrow < URI_j \, P_x \, URI_i >$. Two methods can guarantee the completeness of P_s: One is to store all implicit data brought by P_s. E.g. when user inserts $< URI_i \, P_x \, URI_j >$, both $< URI_i \, P_x \, URI_j >$ and $< URI_j \, P_x \, URI_i >$ will be stored. There is no need to consider P_S character when query with this method. But the volume of such data will be doubled. The other method only stores the explicit data and use query rewriting to satisfy P_S requirement. E.g. suppose $P_x \in P_S$, query $< URI_i \, P_x \, ? >$ should be rewritten as $< URI_i \, P_x \, ? >$ and $<? \, P_x \, URI_i >$. When data volume that has P_S character become larger, performance of the second method will be better than the first one.

3. Rule $P_i \prec P_j \wedge < URI_x\, P_j\, URI_y > \Rightarrow < URI_x \quad P_i \quad URI_y >$makes R_p may bring implicit data. Considering query $< URI_x\, P_i\, ? >$, if there is only $< URI_x\, P_j\, URI_y >$ in R_I, no result will be returned if don't use rule above. As we have mentioned in section 4.1, relation $P_i \prec P_j$ in R_p is stored as a tree structure in HStar. For any P_i in this structure, all its descendants can be accessed directly. So when processing query $< URI_x\, P_i\, ? >$, HStar will search all data in D which have P_i or P_i's descendants as their Property. Special storage design in HStar makes such operation can be processed efficiently. We will give detailed analysis in section 5. Relation $\{Pi \leftrightarrow P_j | < P_i\, \text{owl:inverseOf}\, P_j > \in D\}$, which is defined in R_p, has rule $< P_i\, \text{owl:inverseOf}\, P_j > \wedge < URI_x\, P_i\, URI_y > \Rightarrow < URI_y\, P_j\, URI_x >$ defined in OWL semantic. Like P_s, completely materializing implicit data brought by this rule will double such data volume. Query rewriting can also be used here and its performance will be better when data volume is larger.

4.3 Completeness Analysis of R_{CI}

R_{CI} describes type information of URI and it is the most complex part of OWL data. Both R_C and R_{CP} affect completeness of R_{CI} and the related rules are list below:

1. $< URI_x$ rdf:type $C_j > \wedge C_i \prec C_j \Rightarrow < URI_x$ rdf:type $C_i >$
2. $< P_x$ rdfs:domain $C_y > \wedge < URI_i P_x URI_j > \Rightarrow < URI_i$ rdf:type $C_y >$
3. $< P_x$ rdfs:range $C_y > \wedge < URI_i P_x URI_j > \Rightarrow < URI_j$ rdf:type $C_y >$

As we have mentioned in section 4.1, relation $C_i \prec C_j$ is stored as a tree structure in HStar. When processing query $< URI_x$ rdf:type $? >$, we first get C_i if there is explicit data $< URI_x$ rdf:type $C_i >$ in D; then get all ancestors of C_i and return them as the result. For rules 2 and 3, if we don't get complete R_{CI} relation when loading OWL documents, the whole data space search will be required when query processing. HStar materializes all implicit data brought by rules 2 and 3.

From discussion above, we can observe that HStar only materializes implicit data brought by P_F and P_{IF}, implicit data in R_{CI} brought by Property's domain and range.

5 Storage Design

From the third section, we can see that the main part of OWL data is five kinds of relations, R_C, R_P, R_{CP}, R_I and R_{CI}. How to organize these relations on hard disk is the task of storage design. Considering the characteristics of both OWL data and queries against it, we designed a special storage model for OWL data, which is built on file system rather than RDB, ORDB and etc. In the rest of this section, we will first describe the inner identifier of entities, and then present the storage method of different relations.

5.1 Inner Identifier for Entities: OID

In OWL data, entities are identified by URI, which is usually a long string. Storing original URI takes considerable space; therefore we use inner identifier OID to replace

URI in storage. OID consists of two members: *id*, which occupies four bytes, *flag*, which occupies one byte, indicates whether entity has equivalent resources. Thus an OID totally occupies five bytes, which is much smaller than a URI. The relationship between OID and URI is one to one and is saved in two global hash tables.

5.2 Storage of R_C and R_{CI}

Inheritance relation in R_C is stored in tree structure. We named it C-Tree. E.g. the relation in fig.2 is stored as C-Tree structure in fig.3. Each tree node keeps addresses (represented by page number and offset in physical page) of its first child, parent, left and right siblings. It is easy to access the ancestors and descendents of a tree node by these addresses.

Non-tree nodes in inheritance relation graph split into multiple copies. One is primary (P-Node), and the others are references. E.g. node C_m has been divided into C_{m1}, C_{m2}. They are linked in the Same Entity List (SE-List), with the primary one as head. Only primary node stores the address of child and Individual List.

Locating arbitrary C_x in C-Tree structure is an indispensable operation for inheritance relation query. C-index is built to improve the performance of this operation, which is a B+ tree structure and uses identifiers of P-Nodes as keys. As showed in fig.3, C-index record addresses of nodes in C-Tree. Using C-index and SE-List, all the nodes responding to C_x in C-Tree can be accessed quickly.

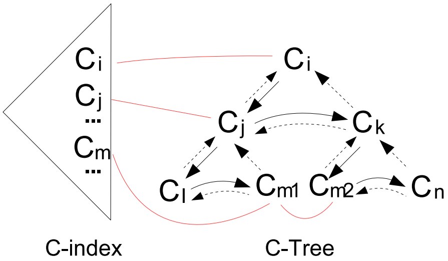


Fig. 3. C-Tree and C-Index for R_C Storage

Equivalence relation in R_C is stored in B+ tree. E.g. suppose C_i is equivalent to C_j and the id of C_i's OID is smaller than the id of C_j's OID, and then take C_i as key and C_j as value. Only explicit equivalence relations are stored. Equivalence sets are built in memory to facilitate query processing, each set corresponding to a memory list. Updating equivalence relation needs to maintain both B+ tree and lists in memory.

Individuals related to the same C_x are stored in one Individual List (I-List), whose start address is saved in C_x's P-Node of C-Tree. E.g. in fig.4, individuals I_i and I_j have type of C_m. They are stored in an I-List, with the start address kept in node C_{m1} of C-Tree. This structure is to facilitate querying individuals of given Class, which is the most frequent query about R_{CI}.

Queries for type of given individual are less frequent but necessary. IC-index is built to facilitate these queries. It is a B+ tree index, which uses OID of individual as key.

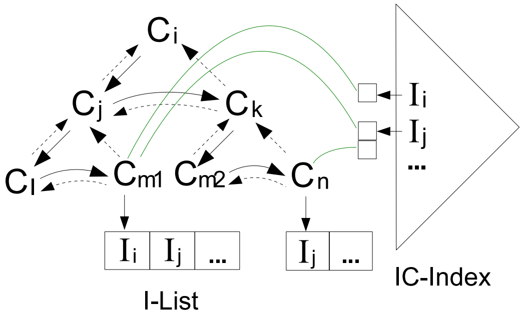


Fig. 4. I-List and IC-index for R_{CI} Storage

Leaf node contains all the Classes to which the individual belongs. E.g. IC-index in fig.4 records that individual I_i belongs to C_m and I_j belongs to C_m and C_n.

Only explicit R_{CI} relations are stored in I-List and IC-index. To guarantee the inference completeness, we need to combine I-List and IC-index with C-Tree structure. That is the reason why we store addresses of I-Lists in P-Nodes of C-Tree. E.g. in fig.4, to find individuals of C_i, I-List of both C_i and its descendants need to be returned. Here, I_i, I_j is the result. To query type of I_i, we find C_{m1} through IC-index, then C_m and its ancestors are returned. Here, C_i, C_j, C_k, C_m is the result.

5.3 Storage of R_p, R_{CP}, R_I and T_p

Inheritance relation and equivalence relation in R_p are stored in the same way as those relations in R_C. P-Tree, and P-index are built as C-Tree and C-index. Inverse relations in R_p are stored as data members of P-Nodes in P-Tree (Property Tree); for $P_i \leftrightarrow P_j$, store P_j in P_i's P-Node, and store P_i in P_j's P-Node.

T_p and R_{CP} are also stored as data members of P-Nodes. T_p is represented by a byte and the first four bits are used to indicate whether P_x has transitive, symmetric, function and inverse-function characters. R_{CP} is stored as two arrays, which store entities having $rdfs : domain$ or $rdfs : range$ relation with P_x.

Equivalence relation in R_I (namely R_{I2} in section 4.2) is stored as same as that relation in R_C. Individual pairs of R_{I1}, which are related to same transitive P_x, are stored in one Individual Tree (I-Tree). I-Tree adopts the same structure as C-Tree, including I-index and SE-List structures. E.g. in fig.5, P_n is transitive. Pairs (I_k, I_m), (I_k, I_n) relate to P_n, and are stored in its I-Tree. Individual pairs related to same non-transitive P_x are stored in two Individual B+ trees (IB-Tree). One is SB-Tree (S-Key B+ tree), taking subject as key. The other is OB-Tree (O-Key B+ tree), taking object as key. E.g. in fig.5, P_m is not transitive. Pair (I_i, I_j) relates to P_m and is stored in its IB-Trees. SB-Tree takes Ii as key and OB-Tree takes I_j as key. The root addresses of I-Tree and IB-Trees are kept in P_x's P-Node.

IP-index is built similarly to IC-index. The difference is that IP-index records how an individual relates to different properties (as subject or object). E.g. the IP-index in fig.5 records that I_i relates to P_m as subject (represented by solid lines), I_j relates to P_m as object (represented by dashed line), and so on.

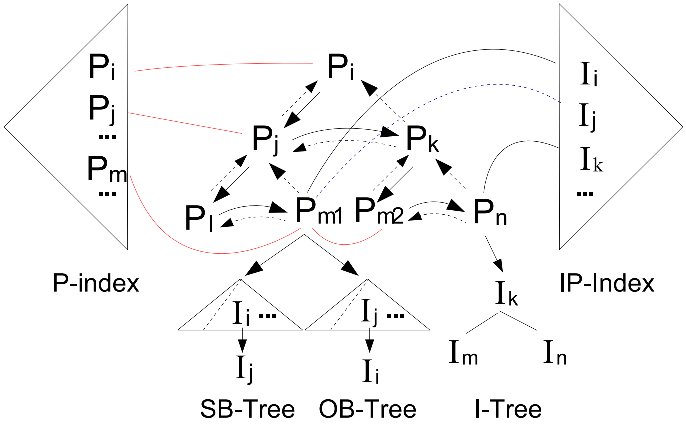


Fig. 5. Storage of R_p and R_{I1}

Queries against R_{I1} can be processed in a similar way with R_{CI}. The difference is that queries against R_{I1} may need further search in P_x's I-Tree or IB-Tree. For transitive P_x, search in I-Tree in the same way as in C-Tree. For non-transitive P_x, search in SB-Tree with given subject, or in OB-Tree with given object.

6 Query Processing

HStar supports queries in SPARQL language, which is proposed by W3C and likely to be the standard query language for OWL. When we mention "OWL query" later, it means SPARQL query. Here we give a query example, which queries all the facts related to "students take courses".

```
PREFIX p:<http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl>
SELECT ?x ?y
WHERE {
       ?y rdf:type p:Course.
       ?x rdf:type p:Student.
       ?x p:takesCourse ?y
}
```

We call triple with variable(s) “Query Triple”, QT for short. E.g. “?y rdf:type p:Course” in the query example is a QT, in which “?y” represents variable to be evaluated during query processing. From query example above, we can observe that QT is the basic unit in OWL query. Query processor first evaluate all QTs to get middle results and then choose some order to join all middle results to get final results. Different join orders produce different sizes of middle results and this affects query performance. Such problem has also been encountered in SQL query processing. For OWL data is different from data in relational database, new solution needs to be proposed. In the next section, we give our intuition for this problem and describe several possible solutions that can be used for OWL query optimization.

6.1 Query Optimization

1. Remove possible redundant QTs based on Ontology.
R_{CP} is part of Ontology and it defines domain and range of a Property. Rules

$$< P_x rdf : domainC_y > \wedge < URI_i P_x URI_j > \Rightarrow < URI_i \text{ rdf:type } C_y > \text{ and}$$
$$< P_x rdf : rangeC_y > \wedge < URI_i P_x URI_j > \Rightarrow < URI_j \text{ rdf:type } C_y >$$

have been mentioned in section 4.3. These rules not only bring implicit data, but also define restrictions. That means if there is $< URI_i P_x URI_j >$ in D and P_x has domain C_m, has range C_n, then URI_i must be an instance of Class C_m and URI_j must be an instance of Class C_n. We can make full use of such restrictions to optimize some type of queries. E.g. suppose there are properties "StudentNumber", "Teach" and two disjoint classes "Student", "Teacher". We know only Class "Student" can have "StudentNumber" and only Class "Teacher" can do "Teach" in real world. These facts will be defined by R_{cp}. Now if user issues query $<?s \text{ StudentNumber} ?n ><?s \text{ Teach} ?c >$,we can immediately judge that such query has no result because "Student" can not "Teach" and "Teacher" has no "StudentNumber". Another example is that if user issues query $<?s \text{ rdf:type Student} ><?s\text{StudentNumber} ?n >$, we can remove QT $<?s \text{ rdf:type Student} >$ because only "Student" has "StudentNumber".

2. Choose Join order based on statistic data.
Choosing join order needs a method to estimate mid-result size of two QTs' join. E.g. query $< s\, p_1\, ?x >, <?x\, p_2\, ?y >, <?y\, p_3\, o >$ contains three QTs. There are two possible join orders: $(< s\, p_1\, ?x >$ join $<?x\, p_2\, ?y >)$ join $<?y\, p_3\, o >$ or $< s\, p_1\, ?x >$ join $(<?x\, p_2\, ?y >$ join $<?y\, p_3\, o >)$. If we can estimate middle results' size of $(< s\, p_1\, ?x >$ join $<?x\, p_2\, ?y >)$ and $(<?x\, p_2\, ?y >$ join $<?y\, p_3\, o >)$, then we can choose the join order which has smaller middle result size. Here we suggest borrowing idea for such problem from relational database. When loading data into HStar, we can compute how many triples there are for every Property, we named this number as N_{tp}; and compute how many different instances there are for every Property's subject and object, we named the two numbers as N_{sp} and N_{op}, then the middle result size of $< s\, p_1\, ?x >$ join $<?x\, p_2\, ?y >$ can be computed by $min\{N_{tp1}/N_{op1}, N_{tp2}/N_{sp2}\}$.

7 Experiment

Experiments in [2] give detailed compare among semantic repositories, DLDB-OWL [6], Sesame-DB [1], Sesame-Memory [1] and OWLJessKB [4]. The experiments test performance of data loading, query processing and query completeness. In our experiment, we do test on systems DLDB-OWL, Sesame-DB, OWLim [2] and HStar. OWLim is a memory-based system, implemented under Sesame general architecture and has better performance on data loading, query processing and query completeness than Sesame's original memory-based system. OWLJessKB [4] is also a memory-based system. [2] points out that it has implemented incorrect inference strategy. So OWLim can be treated as the best memory-based system and we ignore Sesame-Memory and OWLJessKB system in experiment.

Our experiment uses an extension of Lehigh University Benchmark, which has been described in [2]. Four test data sets are generated by tool provided by [2]. They are univer1, univer5, univer10 and univer20. The smallest data set is 8MB including 15 OWL documents. The largest is 218MB including 402 OWL documents. We get "OutOfMemory" error when loading univer10 into OWLim system. Sesame-DB uses user-defined inference rules and costs about 13 hours to load univer5. "OutOfMemory" error occurred when loading univer20 into HStar for a memory-based hash map is used. This will be improved in the next version. DLDB-OWL costs more than 13 hours to load univer10, but it still doesn't finish loading work, which is different from that discussed in [2]. So we just give out the test result for first three data sets.

7.1 Experiment Environment

Hardware: CPU P4.3G, 512MB of RAM, 40GB of hard disk; Software: Windows XP, Java JDK1.5, MySQL4.1.4, MS Access2003, DLDB-OWL(04-03-29 release), Sesame(1.2.2), OWLim(2.8). For all test systems, we set maximum heap size as 256MB.

Table 1. Description of test data sets and data loading performance

	Data set	Instance number	Load time(ms)	Repository size(KB)
OWLim	LUBM(1, 0)	103,074	2,985	17,311
Sesame-DB			1,206,141	48,333
HStar			98,641	19,922
DLDB-OWL			183,937	15,876
OWLim	LUBM(5, 0)	645,649	47,578	107,809
Sesame-DB			47,131,655	283,967
HStar			982,875	77,082
DLDB-OWL			994,157	89,156
OWLim	LUBM(10,0)	1,316,322	-	-
Sesame-DB			-	-
HStar			2,135,453	154,656
DLDB-OWL			-	-

From left of fig.6, we can observe that OWLim has the best data loading performance for the first two data sets. HStar has almost the same performance with DLDB-OWL. Sesame-DB has the worst performance. [2] pointes out that Sesame-DB constructs dependent relation among OWL data elements when loading data. This is very time consumed but is very useful for update performance. DLDB-OWL doesn't consider update problem. HStar just materialize a little part of inference data and it's easy to maintain their relation.

[2] gives 14 query test cases. They are used to test query performance and query completeness. In our experiment, OWLim supports the most semantic rules and we use OWLim query answers as benchmark to evaluate other systems' query completeness.

From fig.7, we can observe that HStar has different answers with OWLim only for the 12th query. DLDB-OWL has no answers for the 11, 12, 13th queries. Sesame-DB has incompleteness answers for the 6, 7, 8, 9th queries and has no answers for the 10, 12th queries. We can sort them by answer completeness as below: OWLim > HStar > DLDB-OWL > Sesame-DB.

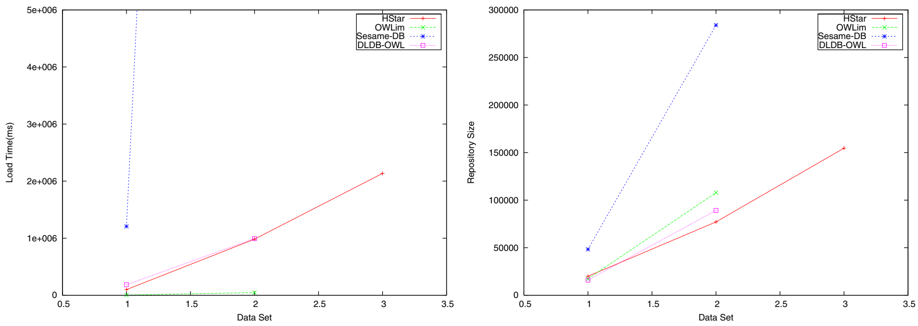


Fig. 6. Data loading performance and repository size

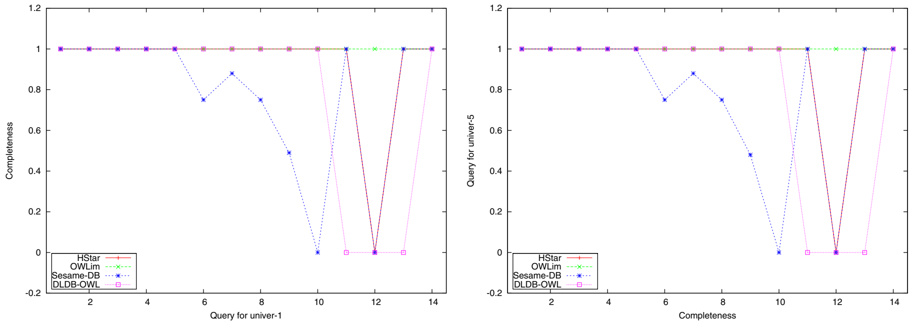


Fig. 7. Query completeness

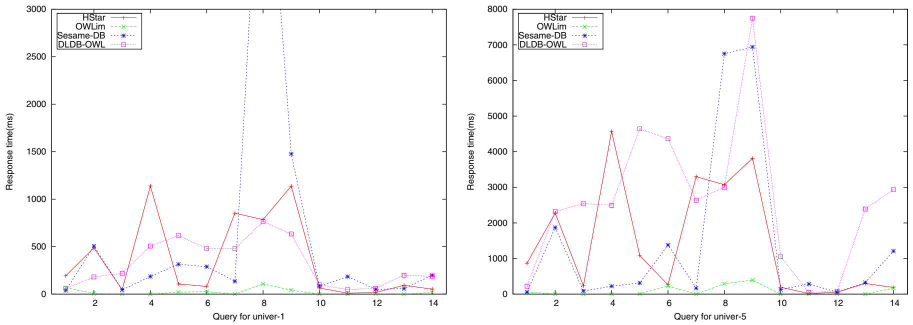


Fig. 8. Query response time

Fig.8 describes the query response time for 14 queries. To avoid impact of OS buffer, we test 10 times for every query and compute the average time. Only OWLim is memory-based, so it has the best query performance. HStar, DLDB-OWL and Sesame-DB have different query process strategies, so they have owned preponderance for different queries.

E.g. HStar has better performance for queries 6, 8, 10, 11, 12, 14 than DLDB-OWL and Sesame-DB, has better performance for query 3 than DLDB-OWL but worse than Sesame-DB, has worse performance for queries 1, 4, 7 than DLDB-OWL and Sesame-DB, has better performance for query 2 than Sesame-DB but worse than DLDB-OWL. For queries 5, 9 and 13, the performance is related with data sets.

From experiments described above, we can summarize that HStar has an ideal performance for data loading, query processing and provides the highest query complete-ness among all hard disk based systems.

8 Conclusion

This paper introduced a semantic repository system called HStar, which is based on file system. We first formalized OWL data model supported by HStar and then gave detailed discussion for completeness problem of OWL data, gave detailed discussion of storage design on file system and query processing strategy. At last, we used exten-sional Lehigh University Benchmark to test HStar and compared it with DLDB-OWL, Sesame-DB, which use relational database, and OWLim, which is memory-based. From experiment, we observed that HStar has an ideal performance for data loading, query processing and provides the highest query completeness among all hard disk based systems. Because HStar has used a memory-based hash map module, ”OutOfMemory” error occurred when loading data set univer20. We plan to design a hard disk based hash structure to replace it in next version of HStar.

Acknowledgments

We would like to thank Dr. Li Ma and Dr. Yue Pan for their constructive suggestions. This work is sponsored by IBM China Research Lab, and is partially supported by the grants from the Natural Science Foundation of China under grant number 60573091, 60273018; the National 973 Basic Research Program of China under Grant No.2003CB- 317000; the Key Project of Ministry of Education of China under Grant No.03044 ; Program for New Century Excellent Talents in University(NCET).

References

1. Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen. Sesame: A generic architecture for storing and querying rdf and rdf schema. In Ian Horrocks and James A. Hendler, editors, *International Semantic Web Conference*, volume 2342 of *Lecture Notes in Computer Science*, pages 54–68. Springer, 2002.
2. Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. An evaluation of knowledge base systems for large owl datasets. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *International Semantic Web Conference*, volume 3298 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2004.
3. Atanas Kiryakov, Damyan Ognyanov, and Dimitar Manov. Owlim - a pragmatic semantic repository for owl. In Mike Dean, Yuanbo Guo, Woochun Jun, Roland Kaschek, Shonali Krishnaswamy, Zhengxiang Pan, and Quan Z. Sheng, editors, *WISE Workshops*, volume 3807 of *Lecture Notes in Computer Science*, pages 182–192. Springer, 2005.

4. Joseph Kopena and William C. Regli. Damljesskb: A tool for reasoning with the semantic web. *IEEE Intelligent Systems*, 18(3):74–77, 2003.
5. Li Ma, Zhong Su, Yue Pan, Li Zhang, and Tao Liu. Rstar: an rdf storage and query system for enterprise resource management. In David Grossman, Luis Gravano, ChengXiang Zhai, Otthein Herzog, and David A. Evans, editors, *CIKM*, pages 484–491. ACM, 2004.
6. Zhengxiang Pan and Jeff Heflin. Dldb: Extending relational databases to support semantic web queries. In Raphael Volz, Stefan Decker, and Isabel F. Cruz, editors, *PSSS*, volume 89 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

Minerva: A Scalable OWL Ontology Storage and Inference System

Jian Zhou[1], Li Ma[2], Qiaoling Liu[1], Lei Zhang[2], Yong Yu[1], and Yue Pan[2]

[1] APEX Data and Knowledge Management Lab,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University,
200240 Shanghai, China
{priest, lql, yyu}@apex.sjtu.edu.cn
[2] IBM China Research Lab,
100094 Beijing, China
{malli, lzhangl, panyue}@cn.ibm.com

Abstract. With the increasing use of ontologies in Semantic Web and enterprise knowledge management, it is critical to develop scalable and efficient ontology management systems. In this paper, we present Minerva, a storage and inference system for large-scale OWL ontologies on top of relational databases. It aims to meet scalability requirements of real applications and provide practical reasoning capability as well as high query performance. The method combines Description Logic reasoners for the TBox inference with logic rules for the ABox inference. Furthermore, it customizes the database schema based on inference requirements. User queries are answered by directly retrieving materialized results from the back-end database. The effective integration of ontology inference and storage is expected to improve reasoning efficiency, while querying without runtime inference guarantees satisfactory response time. Extensive experiments on University Ontology Benchmark show the high efficiency and scalability of Minerva system.

1 Introduction

The rapid growing information volume in World Wide Web and enterprise intranet makes it difficult to access and maintain the information required by users. Semantic Web, the next generation web, aims to provide easier information access and usability by exploiting machine understandable metadata. In recent years, ontology, which enables a shared, formal, explicit and common description of a domain knowledge, has been recognized to play an important role in Semantic Web and enterprise knowledge management. W3C has recommended two standards for publishing and sharing ontologies on the World Wide Web: RDF/RDFS [1] and OWL [2]. OWL builds on top of RDF/RDFS and adds more vocabularies for describing properties and classes, which improves expressiveness but increases reasoning complexity.

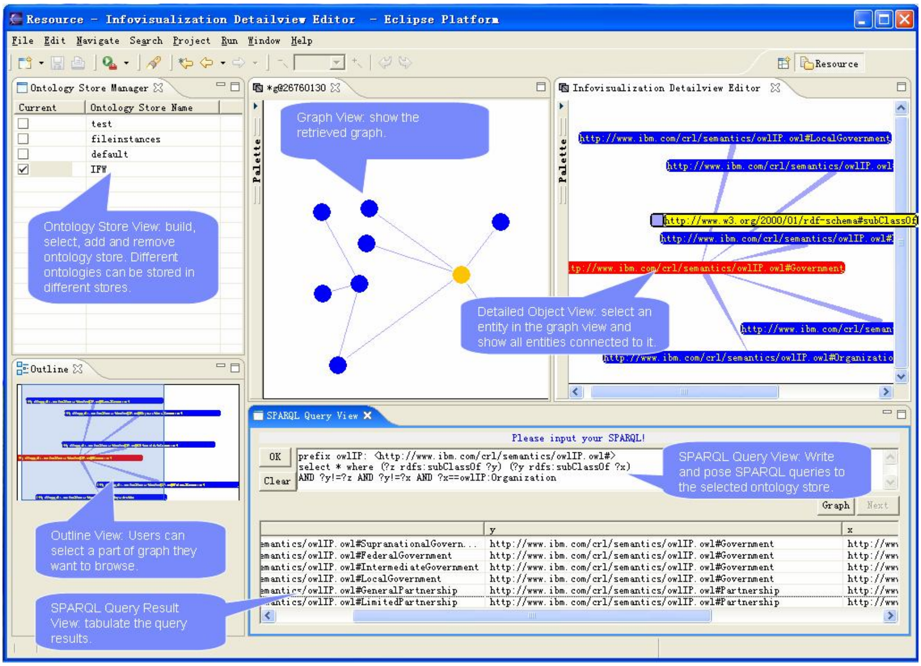
The logical foundation of OWL is Description Logic(DL) [3] which is a decidable fragment of First Order Logic(FOL). Therefore, inference of OWL ontologies

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 429–443, 2006.
© Springer-Verlag Berlin Heidelberg 2006

can be handled by DL reasoners. A DL knowledge base consists of two components, a TBox and an ABox. The TBox describes the terminology, while the ABox contains assertions about individuals. Correspondingly, the DL reasoning includes TBox reasoning(i.e., reasoning with concepts) and ABox reasoning(i.e., reasoning with individuals). It is demonstrated in [4,5] that DL reasoners are able to cope with TBox reasoning of real world ontologies. But the extremely large number of instances of real ontologies makes it difficult for DL reasoners to deal with ABox reasoning. It is critical to develop scalable and efficient ontology management systems.

This paper presents Minerva, a storage, inference and querying system for large-scale OWL ontologies on top of relational databases. It aims to meet scalability requirements of real applications and provide practical reasoning capability as well as high query performance. Minerva is a component of the IBM Integrated Ontology Development Toolkit(IODT) which is publicly available at [6]. Figure 1 shows graphical user interface of Minerva. Using Minerva, one can store multiple large-scale ontologies in different ontology stores, issue SPARQL [7] queries and obtain results listed in tables or visualized as RDF graphs.

In order to achieve high system performance and provide practical inference capability, we combine DL reasoners for the TBox inference with logic rules for the ABox inference. Our method is based on the mapping theory between Description Logic and Logic Programs [8]. It is proved that Description Horn


Resource – Infovisualization Detailview Editor – Eclipse Platform
File Edit Navigate Search Project Run Window Help
Resource
Ontology Store Manager
Current
Ontology Store Name
☐ test
☐ fileinstances
☐ default
☑ IFW
*g@26760130
Palette
Infovisualization Detailview Editor
http://www.ibm.com/crl/semantics/owlIP.owl#LocalGovernment
http://www.ibm.com/crl/semantics/owlIP.owl
http://www.w3.org/2000/01/rdf-schema#subClassOf
http://www.ibm.com/crl/semantics/owlIP.owl#
tp://www.ibm.com/crl/semantics/owlIP.owl#Government
http://www.ibm.com/crl/seman
http://www.ibm.com/crl/semantics/owlIP.owl#Organizatio
Outline
SPARQL Query View
Please input your SPARQL!
OK
Clear
Graph
Next

Graph View: show the retrieved graph.

Ontology Store View: build, select, add and remove ontology store. Different ontologies can be stored in different stores.

Detailed Object View: select an entity in the graph view and show all entities connected to it.

SPARQL Query View: Write and pose SPARQL queries to the selected ontology store.

Outline View: Users can select a part of graph they want to browse.

SPARQL Query Result View: tabulate the query results.

```
prefix owlIP: <http://www.ibm.com/crl/semantics/owlIP.owl#>
select * where (?z rdfs:subClassOf ?y) (?y rdfs:subClassOf ?x)
AND ?y!=?z AND ?y!=?x AND ?x==owlIP:Organization
```

	y	x
emantics/owlIP.owl#SupranationalGovern...	http://www.ibm.com/crl/semantics/owlIP.owl#Government	http://www
emantics/owlIP.owl#FederalGovernment	http://www.ibm.com/crl/semantics/owlIP.owl#Government	http://www
emantics/owlIP.owl#IntermediateGovernment	http://www.ibm.com/crl/semantics/owlIP.owl#Government	http://www
emantics/owlIP.owl#LocalGovernment	http://www.ibm.com/crl/semantics/owlIP.owl#Government	http://www
emantics/owlIP.owl#GeneralPartnership	http://www.ibm.com/crl/semantics/owlIP.owl#Partnership	http://www
antics/owlIP.owl#LimitedPartnership	http://www.ibm.com/crl/semantics/owlIP.owl#Partnership	http://www

Fig. 1. The Graphical User Interface of Minerva

Logic(DHL) ontologies can be translated into a set of logic programs(i.e., logic rules) without loss of semantics. The TBox precomputation by DL reasoners ensures complete and sound inference on classes and properties within OWL-DL. The logic rules translated from DHL implement practical ABox inference, since DHL covers RDFS semantics (except from the recursive meta model) and most practical OWL semantics [8]. Particularly, we customize the relational database schema based on the translated logic rules for efficient inference. Both the TBox and ABox inference results are materialized in the database so that SPARQL queries can be evaluated efficiently. Extensive experiments on University Ontology Benchmark [9] show the high efficiency and scalability of Minerva system.

The rest of this paper is organized as follows. Section 2 gives an overview of Minerva. Detailed storage design, inference and query processing is described in Section 3. Evaluation and results are reported in Section 4. Related work is discussed in Section 5 and Section 6 concludes the paper.

2 Overview

Figure 2 shows the component diagram of Minerva. Minerva is comprised of Import Module, Inference Module, Storage Module (it is an RDBMS schema) and Query Module.

- Import Module. The import module consists of an OWL parser and two translators. The parser parses OWL documents into an in-memory EODM model(EMF ontology definition metamodel) [6][1], and then the DB translator populates all ABox assertions into the back-end database. The function of the TBox translator is twofold, one is to populate all TBox axioms into a DL reasoner and the other is to obtain inference results from the DL reasoner and insert them into the database.
- Inference Module. A DL reasoner and a rule inference engine compose the inference module. Firstly, the DL reasoner infers complete subsumption relationships between classes and properties. Then, the rule engine conducts ABox inference based on the DLP rules. Currently, the inference rules are implemented using DB SQL statements. Besides our developed structural subsumption algorithm [6], Minerva can use well-known RACER [10] and Pellet [11] for TBox inference via DIG interface.
- Storage Module. It is intended to store both original and inferred assertions by the DL reasoner and the rule inference engine. Since inference and storage are considered as an inseparable component in a complete storage and query system for ontologies, we design a specific RDBMS schema to effectively support ontology inference. Currently, Minerva can take IBM DB2, Derby (http://incubator.apache.org/derby/) and HSQLDB (http://www.hsqldb.org/) as the back-end database.

[1] EODM is an implementation of OMG's Ontology Definition Metamodel (http://www.omg.org/cgi-bin/doc?ad/2003-3-40) on Eclipse Modeling Framework(EMF) (http://www.eclipse.org/emf).

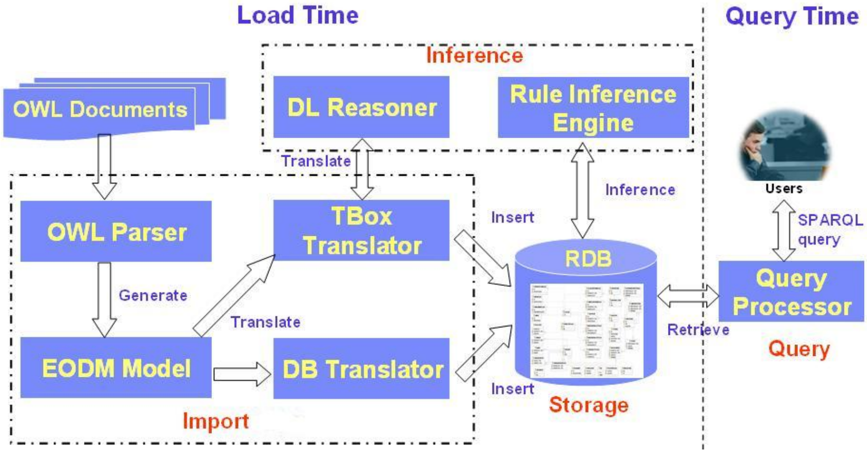


Fig. 2. The Component Diagram of Minerva

- Query Module. The query language supported by Minerva is SPARQL [7]. User SPARQL queries are answered by directly retrieving inferred results from the database using SQL statements. There is no inference during the query answering stage because the inference has already been done at the time of loading data. Such processing is expected to improve the query response time.

In summary, Minerva combines a DL reasoner and a rule engine for ontology inference, materializes all inferred results into a database. The database schema is well designed to effectively support inference and SPARQL queries are answered by direct retrieval from the database. More details about inference and storage are described in next section.

3 Inference, Storage and Querying

3.1 Inference

Grosof et al. [8] defined a new intermediate knowledge representation contained within the intersection between Description Logic(DL) and Logic Programs(LP): Description Logic Programs(DLP), and the closely related Description Horn Logic(DHL). DLP is the LP-correspondent of DHL ruleset. The definition of DHL and DLP makes it practicable to do efficient reasoning of large-scale ontology using the rule inference engine. Considering most real OWL-DL ontologies are more complex than DHL, we extend the original DHL axioms to support OWL-DL-complete[2] TBox inference. More precisely, we use a DL reasoner to obtain all class and property subsumption relationships, instead of supporting

[2] DL reasoners implement sound and complete reasoning algorithms that can effectively handle the DL fragment of OWL.

only DHL axioms. Note that we decompose the complex class descriptions into instantiations of class constructors, assign a new URI to each instantiation and ask the DL reasoner for inference as well. For ABox reasoning, Minerva implements all DLP rules based on the Meta Mapping approach [12]. The Meta Mapping converts all concept and property instances into facts of two predicates `TypeOf` and `Relationship`, and ontology axioms into facts of some predefined predicates(e.g., `SubClassOf` and `SubPropertyOf`). Consequently, there are a fixed number of predefined predicates, reflecting the vocabulary of OWL-DL. Based on these predicates, only a constant rule set is required to cover the semantics of the ontology.

Table 1. The set of rules that cover all DHL axioms. (Rel stands for Relationship, Type stands for TypeOf)

DHL Axioms	Corresponding rule
Rel-Rel Layer(Group 1):	
$P \sqsubseteq Q$	Rel(x, Q, y) :- Rel(x, P, y), SubPropertyOf(P, Q).
$P \equiv Q^-$	Rel(y, Q, x) :- Rel(x, P, y), InversePropertyOf(P, Q).
$P^+ \equiv P$	Rel(x, P, z) :- Rel(x, P, y), Rel(y, P, z), Transitive(P).
$P \equiv P^-$	Rel(y, P, x) :- Rel(x, P, y), Symmetric(P).
Rel-Type Layer(Group 2):	
$\top \sqsubseteq \forall P^-.D$	Type(x, D) :- Rel(x, P, y), Domain(P, D).
$\top \sqsubseteq \forall P.D$	Type(y, D) :- Rel(x, P, y), Range(P, D).
Type-Type Layer(Group 3):	
$C \sqsubseteq D$	Type(x, D) :- Type(x, C), SubClassOf(C, D).
$\exists R.D \sqsubseteq C$	Type(x, C) :- Rel(x, R, y), Type(y, D), SomeValuesFrom(C, R, D).
$C \sqsubseteq \forall R.D$	Type(y, D) :- Rel(x, R, y), Type(x, C), AllValuesFrom(C, R, D).
$D_1 \sqcap D_2 ... \sqcap D_n \sqsubseteq C$	Type(x, C) :- Type(x, D_1), IntersectionMemberOf(D_1, C),..., Type(x, D_n), IntersectionMemberOf(D_n, C).

The DLP rules obtained by the mapping of DHL axioms are listed in Table 1. These rules can be directly handled by deductive databases. Here, we make use of mature relational database to store large-scale ontologies. In order to leverage optimization technologies and scalability of RDBMS as much as possible, we enforce DLP rules using SQL statements on the underlying RDBMS as the implementation of a rule inference engine. [13] shows the semantics of logic programs can be interpreted by the fixed point semantics with respect to Herbrand Models. So we can iteratively execute these rules until no new assertions can be made to obtain the fixed point. The inferred results are materialized in the database so that queries can be evaluated efficiently. Our approach is to trade space for time.

Firstly we use the DL reasoner to calculate the `SubClassOf` relationships between classes and `SubPropertyOf` relationships between properties. The results of this precomputation are stored in the database tables and used by the subsequent rule inference. Rules for ABox inference are categorized into three groups based on their dependency so that rules in group i cannot be fired by rules in group $j (j \geq i)$. This effectively reduces inference costs using SQL statements. Rules in each group will be recursively executed until no new results can be generated. Then the rule engine will proceed to the next group of rules.

The TBox precomputation by DL reasoners ensures complete and sound OWL-DL inference on classes and properties, which can not be covered by DLP rules. For example, if we have axioms $\{Mother \equiv Woman \sqcap \exists hasChild.Person, Parent \equiv Person \sqcap \exists hasChild.Person, Woman \sqsubseteq Person\}$, the implicit relationship that $Mother$ is a subclass of $Parent$ cannot be derived by the DLP rules but will be found by a DL reasoner. After the full TBox reasoning, our rule engine provides complete and sound ABox inference with respect to the semantics of DHL, which covers RDFS semantics (except from the recursive meta model) and most practical OWL semantics [8]. Therefore, our method provides practical inference capability for real applications.

3.2 Storage on Relational Databases

Two best-known ontology toolkits, Jena [14] and Sesame [15], have provided supports for ontology persistent storage on relational database. They persist OWL ontologies as a set of RDF triples and do not consider specific processing for complex class descriptions generated by class constructors(boolean combinators, various kinds of restrictions, etc). [16] proposed to store OWL restrictions in a separate table for ease of representation. However, they did not explain and discuss the effect of their schema on inference in-depth.

The highlight of our database schema is that all predicates in the DLP rules have corresponding tables in the database. Therefore, these rules can be easily translated into sequences of relational algebra operations. For example, rule Type(x, C) :- Rel(x, R, y), Type(y, D), SomeValuesFrom(C, R, D) in Table 1 has four predicates in the head and body, resulting in three tables: `Relationship`, `Typeof` and `SomeValuesFrom`. It is highly straightforward to use SQL statements to execute this rule. We just need to use simple SQL select and join operations among these three tables. The effective integration of ontology inference and storage is expected to significantly reduce inference costs.

We categorize tables of the database schema into 4 types: atomic tables, TBox axiom tables, ABox fact tables and class constructor tables, which are shown in Figures 3 and 4. The atomic tables include: `PrimitiveClass` (in Figure 4), `Property`, `Datatype`, `Individual` and `Literal`. These tables encode the URI with an integer value(the ID column), which reduces the overhead caused by the long URI to a minimum. The hashcode column for URI in `Individual` and `Literal` tables is used to speed up search on individuals and literals. The `Property` table stores URI as well as its characteristics(symmetric, transitive, etc).

There are two important kinds of ABox facts: `TypeOf` and `Relationship`. [17] discussed the advantages of 'Vertical Table'(storing all data in one table with index on the type) in terms of manageability and flexibility to 'Binary Table'(a table for each class and property). Therefore, we adopt 'Vertical Table' to store ABox facts. Tables `SubClassOf`, `SubPropertyOf`, `Domain`, `Range`, `DisjointClass`, `InversePropertyOf` are used to store TBox axioms.

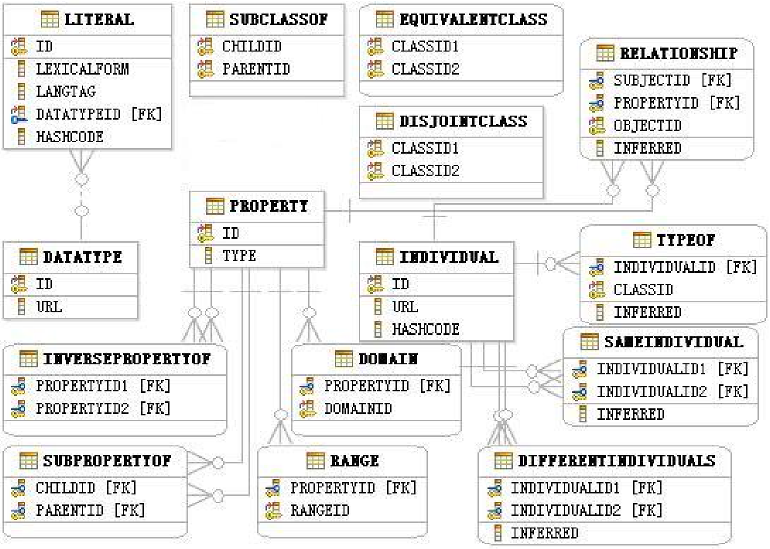


Fig. 3. The relational schema of Atomic, TBox axiom and ABox fact tables

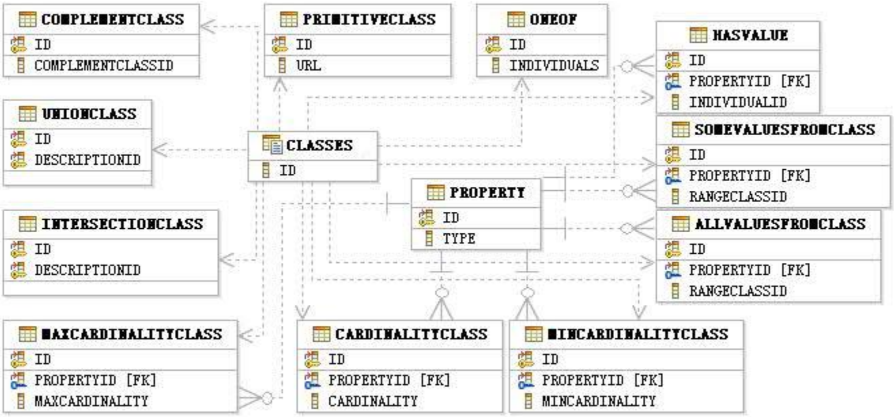


Fig. 4. The relational schema of class constructor tables

The most distinguishing part of our design is class constructor tables in Figure 4. We decompose the complex class descriptions into instantiations of class constructors, assign a new ID to each instantiation and store it in the corresponding class constructor table. Take the axiom $Mother \equiv Woman \sqcap \exists hasChild.Person$ as an example, we first define S_1 for $\exists hasChild.Person$ in `SomeValuesFrom` table. Then I_1 standing for the intersection of $Woman$ and S_1 will be defined in the `IntersectionClass` table. At last, $\{Mother \sqsubseteq I_1, I_1 \sqsubseteq$

Mother} will be added to the `SubClassOf` table. Such a design is motivated by making the semantics of complex class description explicit. In this way, all class nodes in the OWL subsumption tree are materialized in database tables, and rule inference can thus be easier to implement and faster to execute using SQL statements. Also, a view `Classes` is defined to provide an overall view of both named and anonymous classes in OWL ontology.

As introduced in previous section, Minerva materializes all inferred results in the database. Therefore, we have to propose effective methods for ontology update.

1. Addition of TBox Axioms. When new TBox axioms are added, Minerva will send them and the original TBox together to a DL reasoner. Then, we can obtain newly-inferred TBox Axioms and store them into the database. Finally, the rule engine will do ABox inference with only the newly-added and newly-inferred TBox axioms.
2. Addition of ABox assertions. Currently, two kinds of methods for ABox update are supported. The first approach is to add only one assertion at one time. Minerva will determine rules to be fired based on the premise of all ABox rules and obtain inferred assertions using these rules. Then, the inferred assertions are processed one by one in the same manner until no new assertion can be inferred. Another way is relatively straightforward. It just re-runs all ABox inference rules and newly-inferred assertions are materialized into the database.
3. Deletion of TBox Axioms. When some TBox axioms need to be deleted, Minerva will delete all inferred results and redo both TBox and ABox inference, just like populating a new ontology. Obviously, such an update is expensive. But fortunately, ontologies do not change frequently in real applications and thus deletion of TBox Axioms occurs rarely.
4. Deletion of ABox assertions. When deleting an assertion, Minerva first obtains all assertions inferred from this assertion. Then, it runs ABox rules to check whether each of those assertions could be inferred from other existing assertions. By this way, we make sure the safe deletion of an assertion.

3.3 Querying

Recently, W3C has worked out a query language SPARQL [7] for RDF retrieval. The SPARQL query language is based on matching graph patterns. The simplest graph pattern is the triple pattern, which is like an RDF triple but with the possibility of variables in any positions. Minerva has implemented the basic query features of SPARQL, but class expressions are not supported in user queries.

Our query answering algorithm is to simply retrieve results from the database including both original assertions and inferred facts. The query answering module consists of a SPARQL query parser and a SQL translator. In fact, every $x_i : C$ pattern can be translated into a select operation on `TypeOf` table, while every $< x_i, x_j >$: R pattern can be translated into a select operation on `Relationship` table. The translator uses join and union operations on the basic triple selections to build a complete SQL statement and obtain final results. That is, we make effective use of the well-optimized SQL query engine for SPARQL evaluation. This makes Minerva practicable for concurrent queries in various real applications.

4 Evaluation

4.1 Experiment Settings

Experiments are designed to evaluate scalability, efficiency and inference capability of Minerva. We compare our system with OWLIM [18] and DLDB-OWL [19]. These two systems are chosen because it is reported in [20] that DLDB and Sesame(OWLIM is an extension of Sesame) have better performance than other systems in general. DLDB [19] uses the DL reasoner to precompute class and property hierarchies, and employs relational views to answer extensional queries. Its ABox inference mainly supports instance membership reasoning.

Evaluation is conducted on University Ontology Benchmark(UOB) [9], which is extended from the well-known Lehigh University Benchmark(LUBM) [20]. The UOB extends the LUBM in terms of two aspects: 1) include both OWL-Lite and OWL-DL ontologies covering a complete set of OWL-Lite and OWL-DL constructors respectively. 2) add necessary properties to build effective instance links (hence reasoning requirements) and improve instance generation methods accordingly. The UOB consists of university domain ontologies, customizable and repeatable synthetic data, a set of test queries and corresponding answers. In our experiments, we create 3 test sets: OWL Lite-1, OWL Lite-5, OWL Lite-10(The parameter denotes the number of universities). The number of triples is about 220000 for Lite-1, 1100000 for Lite-5 and 2200000 for Lite-10.

There are 13 queries in the benchmark which cover most features of OWL-Lite. The details of all queries can be found in [9]. Here, the evaluation metrics used in [20] are adopted for comparison:

1. *Load time.* The time for storing the benchmark data to the repository, including time for parsing OWL files and reasoning.
2. *Query Response time.* The time for issuing the query, obtaining the result set and traversing the set sequentially.
3. *Completeness and Soundness.* Completeness means the system generates all answers that are entailed by the knowledge base and soundness means all generated answers are correct.

Here, the load time is an average of three times of experiments and the query response time is an average of ten times of experiments. Experiments are conducted on a PC with Pentium IV CPU of 2.66 GHz and 1G memory, running Windows 2000 professional with Sun Java JRE 1.4.2 (JRE 1.5.0 for OWLIM) and Java VM memory of 512M. The version of OWLIM and DLDB we evaluated are v2.8.2 (http://www.ontotext.com/owlim/) and DLDB-OWL (http://swat.cse.lehigh.edu/downloads/dldb-owl.html) respectively.

4.2 Results

Load Time Table 2 compares the load time of three systems. We can see that OWLIM can load the smallest data set OWL Lite-1 using only 29 seconds. It is fastest among these systems. When loading OWL Lite-5 and OWL Lite-10, it reported "Out of Memory" error. In fact, we have also tested other memory-based systems, e.g. RACER [10]. They cannot load the smallest data set OWL Lite-1 which includes about 220,000 triples, because of the memory limitation. The results strongly support our understanding that database technologies should be used to deal with large-scale ontology storage. The average load time of DLDB is less than Minerva's. The difference mainly lies in the inference capabilities and methods of the two systems. DLDB makes use of FaCT for TBox inference and supports a small subset of OWL-Lite in ABox, mainly membership inference based on `SubClassOf` axiom. Minerva implements inference of DLP rules and covers most of OWL-Lite. DLDB constructs views based on inferred class hierarchy information to implement ABox inference, whereas Minerva needs to materialize inferred results by DLP rules in database. Besides additional time for more reasoning, Minerva takes some time to insert inferred results into database. This makes Minerva slower than DLDB to load data. Note that the time in Table 2 includes the reasoning time as these three systems do inference at load time. The time needed for inference in Minerva is about 30%-40% of the load time.

Table 2. Load Time (the unit is second)

	OWL Lite-1	OWL Lite-5	OWL Lite-10
Minerva	868	5469	9337
DLDB	428	1945	3904
OWLIM	29	N/A	N/A

Completeness and Soundness The three systems can answer all queries soundly. That is, they do not return wrong answers for any query. So we only need to check their completeness. Table 3 shows the results. Compared with previous version, OWLIM v2.8.2 can answer all queries correctly. In this new release, more rules are added and inference is made configurable. As is known, OWL-Lite and OWL-DL reasoning cannot be implemented only by rules. That is, OWLIM can conduct only partial OWL-DL TBox inference. This is different from DLDB and Minerva which depend on a DL reasoner for complete TBox inference. Coincidentally, the UOB does not contain a query that needs subsumption inference not covered by existing OWLIM rules. The inference capability of DLDB is relatively weak that it gives 100% complete answers to only 3 queries. Minerva is able to completely process 12 out of 13 queries. Inference on `minCardinality` needed by query 13 is not currently supported in Minerva. As described in Section 3, Minerva makes effective use of DLP rules for ABox inference. Similar to OWLIM, Minerva can add more rules to enhance its ABox inference. In fact, we are currently working on this improvement.

Table 3. Query Completeness (Qi stands for the ith query and the real number denotes $\frac{|Answer_{system} \bigcap Answer_{correct}|}{|Answer_{correct}|}$)

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13
Minerva	1	1	1	1	1	1	1	1	1	1	1	1	0.67
DLDB	1	0.82	1	1	0	0	0	0	0	0.83	0	0.2	0.56
OWLIM	1	1	1	1	1	1	1	1	1	1	1	1	1

Query Response Time Figure 5 shows quantitative comparison on query response time on data sets of different sizes among these systems. The first graph compares performance of three systems on data set OWL Lite-1. OWLIM performs the best in general because its queries are evaluated in memory. Both DLDB and Minerva leverage relational database for query evaluation, which needs expensive IO access to hard disk. On the other hand, DLDB and Minerva have better scalability. This is more or less benefited from the scalability of RDBMS. As OWLIM fails to load other two larger data sets, we do not show its performance curve with the increasing data size. An interesting phenomenon we observed in experiments is about query evaluation of OWLIM. Query 4 includes a four-triple constraint, {?y rdf:type benchmark:Faculty . ?y benchmark:isMemberOf <http://www.Department0.University0.edu> . ?x rdf:type benchmark:Publication . ?x benchmark:publicationAuthor ?y}. If we exchange the order of the 2nd and 3rd triples in the constraint, the response time will increase to 13726ms from only 626ms. That is because OWLIM uses triple patterns in the constraints to filter out irrelevant results. When the most selective triple patterns are at the end of the query, the filtering process would be time-consuming. However, DLDB and Minerva avoid this problem by leveraging query optimization technologies of RDBMS. The second and third graph show the performance of DLDB and Minerva on different data sets. We observed that the query time of Minerva never exceeds 2 seconds, which makes Minerva qualified for practical applications. Also, we found that query response time of Minerva scales well with the data size. The test results of DLDB show that its query time dramatically grows with the increase of the data size and its performance is not as good as Minerva's. DLDB uses class views which is built based on inferred class hierarchy at load time to retrieve instances at query time. DLDB's view query[7] needs to execute union operations in runtime for retrieval. In contrast, Minerva materializes all inferred results, and uses select operations on pre-built index in most cases instead of union operations. This results in less computational costs.

4.3 Discussions

Based on the above results and analysis, we can draw a number of conclusions as well as find that some issues need to be further investigated.

[7] Note that a view is equivalent to a query in relational database.

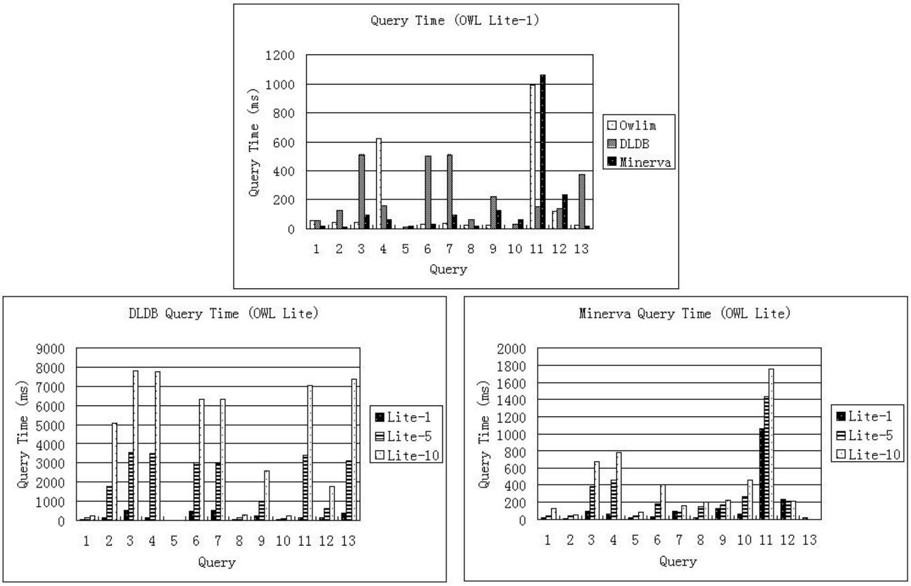


Fig. 5. The average query response time

1. The proposed method for inference is a combination of the DL reasoner and a set of DLP rules corresponding to DHL semantics. It promises that our inference on DHL (a subset of OWL-DL) ontologies is sound and complete as well as that the complete subsumption relationship among classes and properties can be made explicit. Now, we intend to add more rules for ABox inference so that Minerva can handle more expressive ontologies. Also, we are focusing on how to support inference and querying on datatypes (e.g., integers, doubles).
2. Experiments show the high scalability and desirable query optimization of DLDB and Minerva. In fact, this benefits mainly from the underlying relational database. DLDB uses MS Access database and Minerva is built on IBM DB2. In our experiments, we did not change DLDB's back-end database to DB2 as it is reported in [19] that DLDB achieves high performance with default Access database. Further work is to investigate the impact of the underlying RDBMS on the performance of ontology repositories. OWLIM made a significant and meaningful attempt to build native ontology repository and the results are promising. Like the development of native XML storage systems, more efforts are needed for native ontology persistent storage including storage model, query caching and optimization.
3. In Section 3.2, we discussed the ontology update problem. Our method for TBox deletion is expensive though TBox axioms are not deleted frequently. Currently, we are working on an incremental update method for the deletion of TBox axioms.

5 Related Work

Some ontology storage and inference systems have been developed in the past several years. For the sake of efficient storing and querying data with high scalability, there is a trend toward extending RDBMS with OWL inference capabilities, e.g. DLDB [19], Sesame [15], and InstanceStore [21]. Detailed comparisons with DLDB and OWLIM has been reported in previous section.

Sesame is a well-known system which provides efficient storage and expressive querying of large quantities of metadata in RDF/RDFS. In order to support OWL ontology management, Sesame extends its rules. But its simple extension cannot guarantee the inference completeness. OWLIM [18] is another extension for Sesame, which provides a reliable persistence based on N-Triples files. However, its reasoning and query evaluation are performed in memory, which makes it less suitable to handle large numbers of instances in real world ontologies.

InstanceStore [21] implements a restricted form of ABox reasoning on RDBMS. More precisely, it provides sound and complete reasoning on role-free ABox. However, role-free ABox does not include role assertions which describe the relationships between individuals. This guarantees its high efficiency but limits its use in real applications needing role inference.

KAON2 [22] is a successor to the KAON [23] project, an open-source ontology management infrastructure which pays special attention to scalable and efficient reasoning with ontologies. Whereas KAON used a proprietary extension of RDFS, KAON2 is based on OWL-DL and F-Logic. Reasoning in KAON2 is implemented by novel algorithms which reduce a SHIQ(D) knowledge base to a disjunctive datalog program, thus allowing to apply well-known deductive database techniques, such as magic sets or join-order optimizations. ABox assertions can be stored in a relational database (RDBMS), but not all the TBox and ABox inference results are materialized in the database as Minerva.

6 Conclusion and Future Work

This paper presented an RDBMS-based storage and inference system for large-scale OWL ontologies. DL reasoner for the TBox reasoning and rule-based algorithms for the ABox reasoning are combined appropriately. Based on the theoretically proved mapping from Description Logic to Logic Programs [8], we can claim that our system is sound and complete on DHL ontologies. By calculating the subsumption relationship between classes and properties with the DL reasoner, we achieved complete class and property hierarchies and further improved inference capability of Minerva. Extensive experimental results showed the high efficiency and scalability of Minerva.

Acknowledgements

The authors would like to thank GuoTong Xie, Yang Yang, Jing Lu, Sheng-Ping Liu, Lei Li for their helpful discussions and constructive comments, Atanas

Kiryakov and Damyan Ognyanov of OntoText Lab for their great help on evaluation.

References

1. Brickley, D., Guha, R., eds.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation. (2004)
2. Bechhofer, S., van Harmelen, Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A., eds.: OWL Web Ontology Language Reference. W3C Recommendation. (2004)
3. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications., Cambridge University Press (2003)
4. Haarslev, V., Möller, R.: High Performance Reasoning with Very Large Knowledge Bases. In: DL. (2000)
5. Horrocks, I.: FaCT and iFaCT. In: DL. (1999)
6. : IBM's Integrate Ontology Development Toolkit. http://www.alphaworks.ibm.com/tech/semanticstk)
7. Prud'hommeaux, E., Seaborne, A., eds.: SPARQL Query Language for RDF. W3C Working Draft. (2005)
8. Grosof, B.N., Horrocks, I., Volz, R., Decker, S.: Description logic programs: combining logic programs with description logic. In: WWW. (2003) 48–57
9. Ma, L., Yang, Y., Qiu, Z., Xie, G., Pan, Y., Liu, S.: Towards A Complete OWL Ontology Benchmark. In: To appear in European Semantic Web Conference. (2006)
10. Haarslev, V., Möller, R.: RACER System Description. In: Automated Reasoning, First International Joint Conference, IJCAR 2001. (2001)
11. Sirin, E., Parsia, B.: Pellet: An OWL DL Reasoner. In: DL. (2004)
12. Weithöner, T., Liebig, T., Specht, G.: Storing and Querying Ontologies in Logic Databases. In: Proceedings of SWDB'03, The first International Workshop on Semantic Web and Databases, Co-located with VLDB 2003. (2003)
13. Beeri, C.: Logic Programming and Databases. In: ICLP. (1990)
14. Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K.: Jena: implementing the semantic web recommendations. In: (Alternate Track Papers & Posters) WWW. (2004)
15. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In: ISWC. (2002) 54–68
16. Das, S., Chong, E.I., Eadon, G., Srinivasan, J.: Supporting Ontology-Based Semantic matching in RDBMS. In: (e)Proceedings of the Thirtieth International Conference on Very Large Data Bases. (2004)
17. Agrawal, R., Somani, A., Xu, Y.: Storage and Querying of E-Commerce Data. In: VLDB 2001, Proceedings of 27th International Conference on Very Large Data Bases. (2001)
18. Kiryakov, A., Ognyanov, D., Manov, D.: OWLIM - a pragmatic semantic repository for OWL. In: Proceedings of the 2005 International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2005). (2005)
19. Pan, Z., Heflin, J.: DLDB: Extending Relational Databases to Support Semantic Web Queries. In: PSSS1 - Proceedings of the First International Workshop on Practical and Scalable Semantic Systems. (2003)

20. Guo, Y., Pan, Z., Heflin, J.: An Evaluation of Knowledge Base Systems for Large OWL Datasets. In: ISWC. (2004)
21. Horrocks, I., Li, L., Turi, D., Bechhofer, S.: The Instance Store: DL Reasoning with Large Numbers of Individuals. In: DL. (2004)
22. Motik, B., Sattler, U.: Practical DL reasoning over large ABoxes with KAON2, available at http://kaon2.semanticweb.org/. (2006)
23. KAON: (http://kaon.semanticweb.org/)

Exploring the Flexible Workflow Technology to Automate Service Composition

Shuiguang Deng, Ying Li, Haijiang Xia, Jian Wu, and Zhaohui Wu

College of Computer Science, Zhejiang University, Hangzhou 310027, China
{dengsg, wujian2000, haijiangxia, cnliying, wzh}@zju.edu.cn

Abstract. Most of the current workflow-based service composition frameworks and systems require processes to be predefined and services to be statically-bound, thus lacking necessary flexibility to adapt to frequent changes arising from domain/business/user rules and the dynamic Internet environment. This paper proposes a service composition framework based on a flexible workflow method, which enables a part of a process to be created by automatic service composition. In this paper, we propose a semi-automatic service composition framework which enables a part of a process to be created by automatic service composition. In this framework, we encapsulate those uncertain, dynamic and variable parts of a process into black-boxes with a set of rules at the modeling phase. While at the executing phrase, black-boxes are concretized by composing services according to the predefined rules automatically. This framework has been implemented in DartFlow-a service composition platform for the sharing of the TCM (Tradition Chinese Medicine) knowledge and services.

1 Introduction

Service composition has been regarded as an important way to build applications on the fly. At present, there exist a lot of service composition systems and tools based on the workflow technology such as E-Flow [1], SELF-SERV [2], METEOR-S [3] and Active BPEL [4]. They all regard a service composition as a service-oriented workflow including a set of atomic services together with the control and data flow among the services. However, most of them require processes to be predefined and services to be statically-bound. Thus, process designers take up too much time and effort to grasp and draw complex business processes in advance. In our practice using workflow technology to compose services, we confront with many cases in which processes cannot be defined completely in advance but determined according to their execution information. Even though we sketch out the whole processes after considering all possible execution paths, the processes are too complicated to recognize and to manage. Moreover, the predefined processes and statically-bound services are difficult to evolve conveniently according to frequent changes arising from enterprise goals, domain/business/user rules, government policies and the dynamic Internet environment. How to improve the flexibility of service composition to alleviate designers' burden is the issue to be tackled in this study. One possible promising solution is from the AI community which regards the service composition problem as an AI planning problem and proposes various AI panning methods to realize automatic

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 444 – 458, 2006.
© Springer-Verlag Berlin Heidelberg 2006

service composition. The details of service composition as AI planning can be referred to from the surveys written by Peer [5] and Dustdar [6]. Although AI planning methods can generate service compositions automatically according to users' input/output requirements, they do not take the necessary domain/business/user rules into consideration and have no way to ensure the generated service compositions to be in line with the intrinsic core processes of businesses. In fact, on one hand, service compositions are affected by many rules such as domain policies, business constraints and user requirements. While on the other hand, each business process of a service composition has its own fixed core logics needed to be complied with. Furthermore in general, most of business processes needs human beings rather than services to accomplish some activities. Thus it is not applicable for a whole business process to be generated based on automatic service composition by AI planning methods.

In this paper, we propose a service composition framework based on a flexible workflow method to enable a part of a process to be created by automatic service composition. For the fixed core parts of a business process, the framework enables designers to define them in advance and bind services for them statically. While for those dynamic, uncertain and variable parts of a business process, it enables designers to encapsulate them into black-boxes described by rules from domain knowledge, business policies and user requirements. When such a process is put into execution, the black-boxes within the process are concretized by composing services into a subprocess according to the predefined rules automatically. Using this framework to compose services can not only handle complex service compositions but also deal with constant changes arising from domain/business/user rules and the dynamic Internet environment.

2 A Motivating Scenario

In order to make our motivation and proposed solution clear, we illustrate a scenario in TCM (Tradition Chinese Medicine) domain. TCM, as a complete medicine knowledge system, researches into human health conditions via a different approach compared to orthodox medicine [7]. In the DartGrid[1] project [8, 9], which is funded by China Ministry of Science and Technology, we have teamed up with China Academy of Chinese Medical Sciences[2] to realize the sharing of the TCM knowledge and services using the semantic web, grid, web service and workflow technologies.

Let us consider the typical scenario of the TCM clinic diagnosis in which various services are combined to accomplish the whole diagnosis process. Assume that a patient Mary wants to get a diagnosis of her diabetes. Typically, she would firstly log into the local citizen-medical-system and select a preferred hospital and an herbalist doctor. Then she makes a reservation for visiting the doctor. When the reservation time is due, Mary would go to the hospital and register for a new diagnosis. After that, the doctor, Rose in this case, carries out some basic examinations on Mary's body through traditional TCM methods. Now Rose collects enough information about Mary's situation to assess the kinds of Advanced TCM Analysis (ATA in short) Mary

[1] http://ccnt.zju.edu.cn/projects/dartgrid
[2] http://www.catcm.ac.cn/

would need. Rose selects services from a finite set of Advanced TCM Analysis Services (ATAS in short) provided by different TCM organizations according to the hospital's policies and Mary's requirements and composes them into a sub-process according to the TCM-domain rules. After the execution of the sub-process finishes with analysis results returned, Rose makes the final synthetical-diagnosis and prescribes for Mary. Finally, Mary pays for the diagnosis and prescriptions, gets her medicines and ends the whole process.

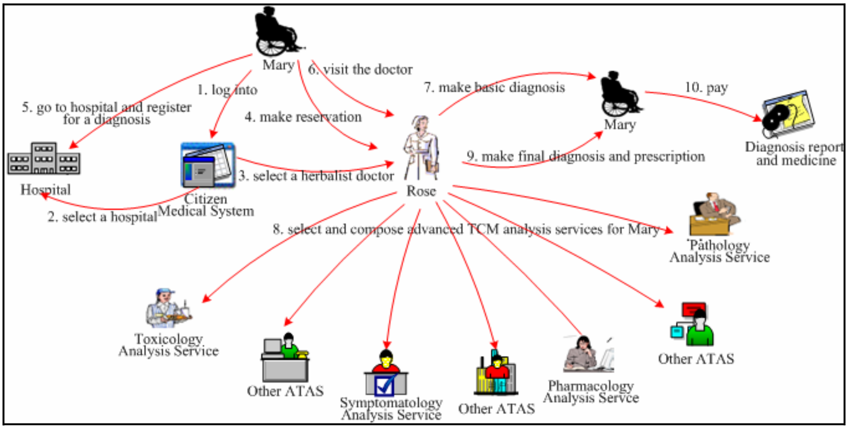

Mary
1. log into
6. visit the doctor
7. make basic diagnosis
10. pay
5. go to hospital and register for a diagnosis
4. make reservation
Mary
Diagnosis report and medicine
Hospital
Citizen Medical System
3. select a herbalist doctor
Rose
9. make final diagnosis and prescription
2. select a hospital
8. select and compose advanced TCM analysis services for Mary
Pathology Analysis Service
Toxicology Analysis Service
Other ATAS
Symptomatology Analysis Service
Other ATAS
Pharmacology Analysis Service
Other ATAS

Fig. 1. A motivation scenario: TCM clinic diagnosis

In the above scenario as Fig.1 shows, we notice that the outline of the whole process is explicit and most of the activities can be predefined. However, the advanced analysis step cannot be defined in advance but depends on Rose's examinations on Mary's situation. That is to say, in this step, which ATA is needed, which ATAS should be selected and how the selected ATAS should be combined are decided dynamically at run time according to the execution information and various policies, rules and requirements. As we have known, there are more than thirty kinds of ATA, such as the toxicology analysis, the symptomatology analysis and the pathology analysis, and among them there are many rules. For example, if both the toxicology analysis and the pharmacology analysis are selected, the pharmacology analysis must precede the other one. Moreover, each kind of ATA has a lot of service providers distributed in over 20 provinces of China. Different selections and compositions of ATAS construct different sub-processes for various patients. There are so many possible selections and combinations for different patients that it is not feasible to construct all the sub-processes in advance and difficult to predefine the whole diagnosis process. In the next section, we propose a framework of service composition to deal with this scenario.

3 A Framework for Service Composition

Workflow is the automation of processes, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules [10]. It has grown to be a major approach to assist the automation of business processes in quite diverse domains. At present, many researchers have coined the term "serviceflow" for service compositions [11, 12], in which services are combined based on business processes to accomplish business goals. In fact, serviceflow can be regarded as a special kind of workflow which includes a set of atomic services together with the control and data flow among the services. The current achievements on workflow modeling, execution and cross-enterprise integration provide the means to compose services in a practicable and convenient way.

In general, utilizing the workflow technology to compose services undergoes two phases: the service composition modeling phase and the execution phase. At the modeling phase, designers build processes according to business logics in a drag-and-drop way within a graphical-style workspace. Each node of the processes is bound to an outer service. While in the execution phase, an execution engine is used to interpret and execute the service compositions by invoking services step by step. Some prototypes and systems [2, 3] have supported services to be bound dynamically in the execution phase in order to improve the flexibility. However, all the current workflow-based service compositions need processes to be predefined. Thus, they are not applicable for many cases such as the aforementioned scenario. In this section, we propose an enhanced service composition framework shown in Fig. 2 for service composition based on a flexible workflow method, which utilizes the "black-box" mechanism to deal with those service compositions which can not be predefined completely.

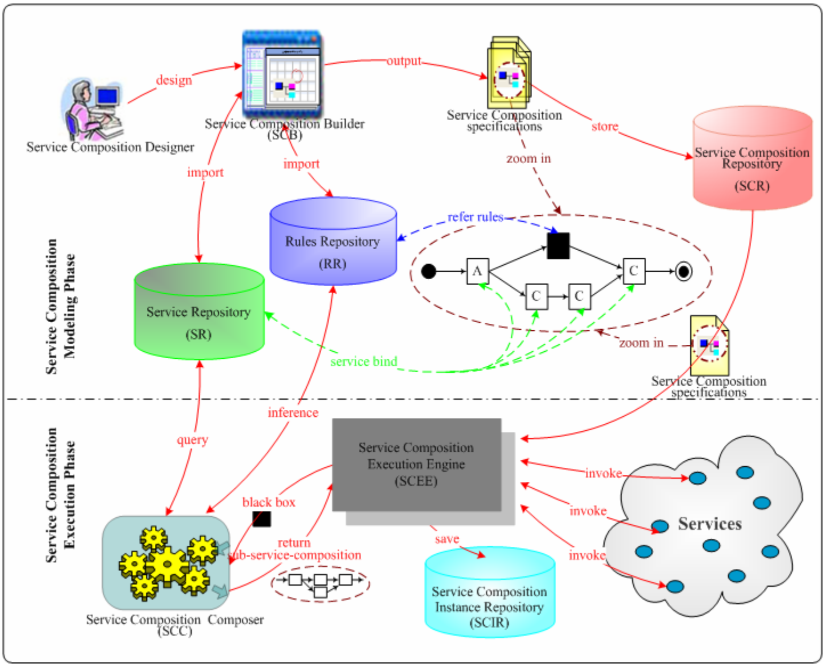

design
output
Service Composition Designer
Service Composition Builder (SCB)
Service Composition specifications
store
Service Composition Repository (SCR)
import
import
zoom in
Rules Repository (RR)
refer rules
A
C
C
C
Service Repository (SR)
service bind
zoom in
Service Composition specifications
Service Composition Modeling Phase
Service Composition Execution Phase
inference
query
Service Composition Execution Engine (SCEE)
invoke
invoke
invoke
Services
black box
return
sub-service-composition
save
Service Composition Composer (SCC)
Service Composition Instance Repository (SCIR)

Fig. 2. A service composition framework based on a flexible workflow method

As Fig. 2 shows, the framework supports both modeling and executing service compositions. It consists of the following main components:

– **Service Composition Builder (SCB):** this component provides an integrated development environment for users to design processes in line with business logics. It enables designers to define the fixed parts of processes in advance and encapsulate the dynamic, uncertain and variable parts into black-boxes. Each black-box is described using rules selected from RR. For example, designers can encapsulate the advanced analysis step of the aforementioned case into a black-box named "advanced analysis black-box" with TCM-domain rules and hospital policies from RR to describe the black-box. In order to ensure the process specification valid, i.e. the process is live, with no deadlock, etc. SCB can utilize formal methods such as Petri-net, process-calculus to verify the process.

– **Service Repository (SR):** This component is responsible for maintaining component services and provides the interfaces for users to advertise and query services.

– **Rules Repository (RR):** this component maintains the domain rules, business rules and user rules. They are used to define black-boxes at the modeling phase and guide how to compose services automatically at the execution phase. For example, one rule may be selected for the advanced analysis black-box in our scenario to indicate that the pharmacology analysis must be selected for a diabetic. The classification and specification of the rules in RR are given in the next section.

– **Service Composition Repository (SCR):** this component is used to maintain service composition specifications and it provides interfaces for SCB to save and load specifications. To avoid starting from scratch, designers can build new compositions based on an existing specification loaded from SCR.

– **Service Composition Execution Engine (SCEE):** it is responsible for interpreting service composition specifications, creating service composition instances, invoking outer services as well as transmitting data among services. When a composition instance runs to one step, SCEE will examine the step first. If the step is a black-box, SCEE transfers it to SCC and waits until SCC returns a sub-service-composition. Otherwise, SCEE invokes the service bounded to the step.

– **Service Composition Composer (SCC):** this component generates a service composition automatically according to the rules associated with the black-box. The details on how SCC works is presented in Section 5.

– **Service Composition Instance Repository (SCIR):** it is used to save the information about service composition instances including instance status and data transferred among participating services.

The target of the framework is to enable services to be composed without a completely-predefined process. The black-box mechanism of the framework enables those uncertain and dynamic factors in service compositions to be determined according to the execution information and predefined rules at the execution phase. Thus, it enhances the flexibility for service compositions to a great extent.

4 Rules Classification and Description

A black-box can be regarded as a container for the uncertain, dynamic and variable parts of service compositions. It must be associated with some rules which will instruct to select atomic services and compose them into a sub-service-composition at the execution phase. Rules can not only be imported from the RR at the building phase, but also be added at the execution phase. In general, rules can be classified into three large categories: domain rules, business rules and user rules.

– **Domain rules** are defined by domain experts according to domain knowledge. In the TCM clinic diagnosis, for example, the pulse analysis service must be selected for a patient with edema. For another example, the pharmacology analysis service must precede the toxicology analysis service if both of them are selected.

– **Business rules** are defined by organizations according to their business behaviors. For example, a city-level TCM hospital may define a rule, stating that the pharmacology analysis service provided by the province-level TCM hospital is preferred than others.

– **User rules** are defined by users who participate in the execution phase. For example, in the aforementioned scenario, Mary may define that the execution time of the advanced analysis black-box must be no longer than 30 minutes and the total cost no more than 150 dollars.

Table 1. Rule classification and rule names

Rule Classification	***Rule Sub-category***	***Rule Name***
Domain Rule	Abstract Service Selection Rule (ASSR)	Choice Rule
		Exclusion Rule
		Condition Rule
		Determination Rule
	Abstract Service Composition Rule (ASCR)	Sequence Rule
		Adjacency Rule
		Data Dependency Rule
Business Rule	Concrete Service Binding Rule (CSBR)	Preference Rule
		Set Confine Rule
		Correlation Rule
User Rule	User QoS Rule (UQR)	Local QoS Rule
		Global QoS Rule

The three classes of rules can be divided into sub-categories as Table 1 shows. Before introducing the details of rules, we give some formal definitions.

Definition **(*abstract-service*).** an abstract service is a 4-tuple: $\psi = (\mu, \sigma, \gamma, \varsigma)$ where:

(1) μ is the name;

(2) σ is the functional description;

(3) $\gamma = \{i_1, i_2, ..., i_m\}$ is the set of the inputs;

(4) $\varsigma = \{o_1, o_2, ..., o_n\}$ is the set of the outputs.

An abstract service represents a functional step rather than a concrete service in a service composition, which is equivalent to the activity concept in workflow. An abstract service will be bound to a concrete service dynamically in the execution phase.

***Definition* (*concrete service*).** a concrete service is an 8-tuple: $\chi = (\eta, \tau, o, \omega, \upsilon, \rho, \varepsilon, \varphi)$ where:

(1) η is the name;
(2) τ is the functional description;
(3) o is the information of the service provider;
(4) $\omega = \{i_1, i_2, ..., i_m\}$ is the set of the inputs;
(5) $\upsilon = \{o_1, o_2, ..., o_n\}$ is the set of the outputs;
(6) ρ is the service precondition;
(7) ε is the service effort;
(8) φ is the information about access point and invoking method.

A concrete service is an existent physical service provided by an outer organization and must be registered into SR as Fig. 2 shows.

***Definition* (*bind-relation*).** a bind-relation is a function mapping an abstract service to a concrete service $ƛ : A = \{\psi_1, \psi_2, ..., \psi_m\} \rightarrow C = \{\chi_1, \chi_2, ..., \chi_n\}$.
For example, $ƛ(\psi)=\chi$ means the concrete service χ is bound to the abstract service ψ.

***Definition* (*black-box*).** a black-box in a service composition is a 3-tuple: $\mho = (\lambda, A, R)$ where:

(1) λ is the name;
(2) $A = \{\psi_1, \psi_2, ..., \psi_m\}$ is the set of abstract services;
(3) $R = \{r_1, r_2, ..., r_n\}$ is the set of rules and each of them is with the general format: $r = (t, Ar, e)$ where:
  a) t is the rule type;
  b) Ar is the set of abstract services referred by the rule;
  c) e is the expression of the rule.

Note that the set A can be empty at the modeling phase, but be filled with abstract services in the execution phase. For example, the doctor Rose will fill the advanced analysis black-box with selected ATA for Mary at the execution time in the aforementioned scenario. On the other hand, the set R is not empty but contains some rules selected from RR shown in Fig. 2. Moreover, users can insert newly-defined rules into R in the execution phase. For example, Mary can insert her QoS rules into R.

– Abstract Service Selection Rule (ASSR)
ASSR defines how to select abstract services into black-boxes. At present, we only consider the following four rules in the TCM diagnosis scenario in this category.

Definition **(*Choice Rule*).** Given a set of abstract services $A = \{\psi_1, \psi_2, ..., \psi_n\}$, a choice rule, denoted as $\psi_1 \oplus \psi_2 \oplus ... \oplus \psi_n$, defines that at least one abstract service $\psi_i \in A(1 \le i \le n)$ must be selected.

Definition **(*Exclusion Rule*).** Given a set of abstract services $A = \{\psi_1, \psi_2, ..., \psi_n\}$, an exclusion rules, denoted as $\psi_1 \otimes \psi_2 \otimes ... \otimes \psi_n$, defines that only one abstract service from A can be selected.

Definition **(*Condition Rule*).** Given two abstract services ψ_1 and ψ_2, an condition rule, denoted as $\psi_1 \rhd \psi_2$, defines that if ψ_1 is selected, ψ_2 must be selected.

Definition **(*Determination Rule*).** Given an abstract service ψ, a determination rule, denoted as $\odot\psi$, defines that ψ must be selected whereas a determination rule, denoted as $\Theta\psi$, defines that ψ cannot be selected.

– Abstract Service Composition Rule (ASCR)
ASCR defines how to combine abstract services together into sub-service-compositions for black-boxes. It will influence the structures of sub-service-service-compositions.

Definition **(*Sequence Rule*).** Given two abstract services ψ_1 and ψ_2, a sequence rule (denoted as $\psi \xrightarrow{*} \psi_2$) defines that ψ_1 must be executed before ψ_2, but need not be adjacent to ψ_2.

Definition **(*Adjacency Rule*).** Given ψ_1 and ψ_2 as in the above definition, an adjacent sequence rule (denoted as $\psi_1 \mapsto \psi_2$) defines that ψ_1 must be adjacent to ψ_2 and be executed before ψ_2.

Definition **(*Data Dependency Rule*).** For two abstract services ψ_1 and ψ_2, a data dependency rule (denoted as $\psi_1 \xrightarrow{D} \psi_2$) defines that all or part of the inputs of ψ_2 come from the outputs of ψ_1. In fact, the rule $\psi_1 \xrightarrow{D} \psi_2$ implies that the rule $\psi_1 \xrightarrow{*} \psi_2$ holds.

Note that we do not consider the parallel rule, which defines two abstract services executed in parallel. This is due to the fact that if there are no composition rules between two abstract services, they can be composed in parallel as default. Thus it is not necessary to define the parallel relationship among abstract services definitely.

– Concrete Service Binding Rule (CSBR)
At the execution phase, an abstract service of a black-box will be bound to a concrete service that is existed in the real world. In general, there are many concrete services that can offer the same functions as the abstract service needs but are provided by different organizations. CSBR is used to instruct to select a proper concrete service from candidates for abstract services mainly according to the business behalf

of organizations. For example, a TCM hospital may establish a serial of rules to instruct doctors and patients to select advanced analysis services. At present, we consider the following rules in this category.

Definition **(*Preference Rule*).** Given an abstract service ψ and a concrete service χ, a preference rule(denoted as $\psi \xleftarrow{\bullet} \chi$) enable the equation $\lambdabar(\psi) = \chi$ hold, which says χ is the preferred selection for ψ.

Definition **(*Set Confine Rule*).** Given an abstract service ψ and a set of concrete services $C = \{\chi_1, \chi_2, ..., \chi_n\}$, a set confine rule (denoted as $\psi \prec C$) enables $\exists \chi \in C \;\; \lambdabar(\psi) = \chi$, which says the service bound to ψ must be selected from C.

Definition **(*Correlation Rule*).** Given two abstract services (ψ_1 and ψ_2) and two concrete services (χ_1 and χ_2), a correlation rule (denoted as $\lambdabar(\psi_1) = \chi_1 \Rightarrow \lambdabar(\psi_2) = \chi_2$), defines that if ψ_1 is bound to χ_1, ψ_2 must be bound to χ_2.

– User QoS Rule (UQR)

Binding concrete services to abstract ones needs to consider not only the above business rules but also user rules. In general, users participating in service compositions, for example, Mary in the scenario mentioned before, may define her acceptable cost for each ATA that she needs. Moreover, Mary can also define her acceptable cost for the whole advanced analysis black-box. At present, we consider the response time and the cost of a service and use the following formula to calculate QoS.

$$QoS(\chi) = T(\chi) \times w + C(\chi) \times (1 - w), \text{ where } 0 \le w \le 1$$

In this formula, $T(\chi)$ and $C(\chi)$ represent the response time and the cost to consume the service χ, respectively, and *w* represent the weight. Note that, the parameter *w* is given by users such as Mary in our scenario. If Mary cares only the cost, she can assign 0 to *w*. User QoS rule can be divided into the local and global QoS rules as follows.

Definition **(*Local QoS Rule*).** Given an abstract service ψ and two numerical values (*l* and *r*, where $l \le r$), a local QoS rule (denoted as $l \le QoS(\lambdabar(\psi)) \le r$), defines that the expected QoS value for ψ ranges from *l* to *r*.

Definition **(*Global QoS Rule*).** Given an black-box $\mho = (\lambda, A, R)$ and the same two values (*l* and *r*) as above, a global QoS rule (denoted as $l \le Qos(\mho) \le r$), defines that the expected QoS value for the whole black-box $\mho$ ranges from *l* to *r*.

In general, the domain rules and business rules are predefined and stored in the component RR in the framework, whereas the user rules are given by participants at the execution phase.

5 Automatic Service Composition Based on Rules

In the framework, SCC accepts a black-box from SCEE and returns a sub-service-composition as the substitute for the black-box. SCC composes services according to the rules associated with a black-box in the following three steps.

– Verification of Abstract Service Selection Rules
The target of this step is to verify whether the selection of abstract services by participants is in line with the predefined abstract service selection rules in the black-box. As our scenario shows, the doctor Rose will assess and select the kinds of ATA that Mary would need. All the doctor needs to do is just to drag-and-drop the target ATA into the advanced analysis black-box for Mary. Because the selection is a manual action, the result of the selection is error prone. Thus, it is necessary to verify the selection before to compose the selected abstracted services. The algorithm *VERIFICATION_SELECTION* is presented below.

```
ALGORITHM : VERIFICATION_SELECTION
INPUT : a black-box $\Omega = (\lambda, A, R)$, where $A = \{\psi_1, \psi_2, ..., \psi_m\}$ and $R = \{r_1, r_2, ..., r_n\}$
OUTPUT : return true if the selection of abstract services is valid; otherwise return false.
METHOD :
  FOR $i = 1$ TO $|R|$ DO
    IF $r_i.t =$ "Choice Rule" AND $|r_i.Ar \cap A| < 1$ THEN
      RETURN false;
    IF $r_i.t =$ "Exclusion Rule" AND $|r_i.Ar \cap A| > 1$ THEN
      RETURN false;
    IF $r_i.t =$ "Condition Rule" AND $r_i.e = \psi_j \triangleright \psi_k$ THEN
      IF $\psi_j \in A$ AND $\psi_k \notin A$ THEN
        RETURN false;
    IF $r_i.t =$ "Determination Rule" THEN
      IF $r_i.e = \odot\psi$ AND $\psi \notin A$ THEN
        RETURN false;
      IF $r_i.e = \Theta\psi$ AND $\psi \in A$ THEN
        RETURN false;
  RETURN true;
```

– Compose Abstract Services
Abstracted services are composed into a sub-service-composition based on rules associated with the black-box. The definition of sub-service-composition is given below.

Definition (Sub-service-composition). A sub-service-composition is a directed acyclic graph, denoted as a 4-tupe: $G = (N, A, C, E)$ where:

(1) $N = \{start, end\}$ is the set of control node containing two elements "*start*" and "*end*", which represent the starting node and end node in the graph, respectively.

(2) $A = \{\psi_1, \psi_2, ..., \psi_n\}$ is the set of abstract services and each of them is a node in *G*.

(3) $C = \{\chi_1, \chi_2, ..., \chi_n\}$ is the set of concrete services and satisfy the following relation: $\forall \psi \in A, \exists \chi \in C, \lambda(\psi) = \chi$

(4) $E \subseteq N \times A \cup A \times N \cup A \times A$ is the set of directed edges and each edge connects an ordered pair of vertices $< v, w >$ where $v \neq w$, v is the tail of the edge and w is the head of the edge.

Using the *ABSTRACT_SERVICE_COMPOSITION* algorithm to compose abstract services needs to keep one principle in mind that if there are no composition rules defined between two abstract services, they can be composed in parallel as default. If the algorithm returns *NULL*, it indicates there are conflicts in the abstract selection rules. How to detect the conflicts among rules is our future direction.

```
ALGORITHM : ABSTRACT _ SERVICE _ COMPOSITION
INPUT : a black - box $℧ = (\lambda, A, R)$, where $A = \{\psi_1, \psi_2, ..., \psi_m\}$ and $R = \{r_1, r_2, ..., r_n\}$
OUTPUT : an abstract sub - service - composition $G(N, A, \phi, E)$.
METHOD :
  Initialize $N = \{start, end\}$;
  Set $RA = \phi$; $C = \phi$;
  FOR i = 1 TO $|R|$ DO
    IF ($r_i.t$ = "Sequence Rule" AND $r_i.e = \psi_j \xrightarrow{*} \psi_k$) AND $|\, r_i.Ar \cap A| = 2$ THEN
      draw an edge $< \psi_j, \psi_k >$ and add it into E; add $\psi_j$ and $\psi_k$ into RA;
    IF ($r_i.t$ = "Adjacency Rule" AND $r_i.e = \psi_m \mapsto \psi_n$) AND $|\, r_i.Ar \cap A| = 2$ THEN
      draw an edge $< \psi_m, \psi_n >$ and add it into E; add $\psi_j$ and $\psi_k$ into RA;
    IF ($r_i.t$ = "Data Dependency Rule" AND $r_i.e = \psi_p \xrightarrow{D} \psi_q$) AND $|\, r_i.Ar \cap A| = 2$ THEN
      draw an edge $< \psi_p, \psi_q >$ and add it into E; add $\psi_j$ and $\psi_k$ into RA;
  For i = 1 TO $|A|$ DO
    IF $\psi_i \notin RA$ THEN
      draw two edges $< start, \psi_i >$ and $< \psi_i, end >$ and add them into E;
    IF there are no edges emitted from $\psi_i$ THEN
      draw an edge $< \psi_i, end >$ and add them into E;
    IF there are no edges entering $\psi_i$ THEN
      draw an edge $< start, \psi_i >$ and add them into E;
  Check whether there are cycles in G through top Topological Sort
  IF there are no cycles in G
    RETURN $G(N, A, \phi, E)$;
  ELSE
    RETURN NULL
```

```
$ALGORITHM : BIND\_SERVICE$
$INPUT$ : $a\ black\text{-}box\ \mho = (\lambda, A, R),\ where\ A = \{\psi_1, \psi_2, ..., \psi_m\}\ and\ R = \{r_1, r_2, ..., r_n\};$
        $an\ abstract\ sub\text{-}service\ composition\ G = (N, A, \phi, E)\ of\ \mho,\ where\ G.A = \mho.A;$
        $a\ set\ of\ registered\ concrete\ service\ S;$
$OUTPUT$ : $return\ a\ concrete\ sub\text{-}service\text{-}composition\ G(N, A, C, E)$
          $if\ there\ exists\ G\ which\ complies\ with\ R;\ Otherwise,\ return\ NULL;$
$METHOD$ :
  $Set\ RA = \phi, C = \phi;$
  $FOR\ i = 1\ TO\ |R|\ DO$
    $IF\ r_i.t = \text{"Preference Rule"}\ AND\ r_i.e = \psi_j \overset{\bullet}{\longleftarrow} \chi$
    $AND\ |r_i.Ar \cap A| = 1\ AND\ \chi \in S\ \ THEN$
      $set\ ƛ(\psi_j) = \chi;\ add\ \chi\ to\ C; add\ \psi_j\ to\ RA;$
    $IF\ \ r_i.t = \text{"Set Confine Rule"}\ AND\ r_i.e = (\psi_j \prec RC = \{\chi_1, \chi_2, ..., \chi_p\})$
        $AND\ |r_i.Ar \cap A| = 1\ THEN$
      $select\ \chi\ with\ the\ minimum\ Qos\ value\ from\ RC,$
        $i.e.\ QoS(\chi) = Min(QoS(\chi_1), QoS(\chi_2), ..., QoS(\chi_p))$
      $IF\ \chi \in S\ \ THEN$
        $set\ ƛ(\psi_j) = \chi; add\ \chi\ to\ C; add\ \psi_j\ to\ RA;$
  $FOR\ i = 1\ TO\ |A|\ DO$
    $IF\ \psi_i \notin RA\ THEN$
      $select\ \chi\ with\ the\ minimum\ Qos\ value\ from\ the\ candidate\ concrete\ services\ for\ \psi_i;$
      $set\ ƛ(\psi_i) = \chi; add\ \chi\ to\ C;$
  $FOR\ i = 1\ TO\ |R|\ DO$
    $IF\ r_i.t = \text{"Correlation Rule"}\ AND\ r_i.e = (ƛ(\psi_\alpha) = \chi_\lambda \Rightarrow ƛ(\psi_\beta) = \chi_\gamma)$
    $AND\ |r_i.Ar \cap A| = 2\ THEN$
      $IF\ \ ƛ(\psi_\alpha) = \chi_\lambda\ AND\ ƛ(\psi_\beta) \neq \chi_\gamma\ in\ \ G\ THEN$
        $remove\ ƛ(\psi_\beta)\ from\ C;\ set\ ƛ(\psi_\beta) = \chi_\gamma;\ add\ \chi_\gamma\ into\ C;$
  $FOR\ \ i = 1\ TO\ |R|\ DO$
    $IF\ r_i.t = \text{"Local QoS Rule"}\ AND\ r_i.e = (l \le QoS(ƛ(\psi_j)) \le r)\ AND\ |r_i.Ar \cap A| = 1\ THEN$
      $IF\ QoS(ƛ(\psi_j)) > b\ OR\ QoS(ƛ(\psi_j)) < l\ THEN$
        $RETURN\ NULL;$
    $IF\ r_i.t = \text{"Global QoS Rule"}\ AND\ r_i.e = (l \le QoS(\mho) \le r)\ AND\ |r_i.Ar \cap A| = 1\ THEN$
      $IF\ QoS(G) > b\ OR\ QoS(G) < l\ THEN$
        $RETURN\ NULL;$
  $RETURN\ G(N, A, C, E)$
```

– Bind Concrete Services

If the *ABSTRACT_SERVICE_COMPOSITION* does not return *NLLL*, an abstract sub-service-composition is generated for a black-box. In this step, the *SERVICE_BIND* algorithm binds each abstract service to a concrete service and generates the final concrete sub-service-composition for the black-box. This algorithm firstly binds concrete services for the abstract services which are referred by CSBR except the Correlation Rules, and then binds concrete services for other abstract ones. After that, it modifies the binding according to the Correlation Rule. At last, it verifies whether the concrete sub-service-composition satisfies the UQR.

6 Related Work

Using workflow methods to develop and manage service compositions is an intuitive way. There are already a good body of projects and work to make the workflow technology more adaptive and convenient for developing and managing service compositions [1-3, 11, 13, 17, 18, 19, 20]. Some focuses on adapting the workflow modeling, composition and verification methods to specify and verify service compositions [13, 17, 19, 20]; some focus on using the workflow execution and monitoring methods to run and manage service compositions [11, 2, 18]. Due to the limitation space, we only introduce some typical work here.

METEOR-S [3] is a project initiated by the LSDIS Lab at the University of Georgia to build a framework for semantic web process composition using the technologies from workflow, semantic web and service areas. It characterizes the use of semantics in service description, registration, composition and execution. Although it uses semantic service templates and semantic process templates to facilitate the composition, it requires the web processes predefined completely. Thus, it does not support dynamic service composition at the execution phase and can not be applied to those kind of cases mentioned in our scenario.

SELF-SERV [2] is a platform proposed by the SOC group at the University of New South Wales using the workflow technology to realize web services composed and executed in a peer-to-peer environment. It uses the state chart to model a service composition and adopts the concept of "service community" to support services to be later-bound. The characteristic of this platform is to enable compositions executed in a P2P fashion with the help of peer software components for each constituent web service. But it has the same shortages as METEOR-S.

E-FLOW [1] is a framework provided by Hewlett-Packard for developing and managing composite e-services, which has a same target as the one of our framework. It divides service nodes within a composition into ordinary service nodes and generic service node. The generic node approach, which is something like our black-box mechanism, supports dynamic process definitions and provides considerable flexibility and adaptability in changing environment. But it needs too much manual participation to concretize generic nodes at the execution phase and does not support the automatic generation of compositions for generic nodes. Moreover, it has no way to ensure the correctness of the outcome of the concretization.

Shazia Sadiq [21] introduces the notion of an open workflow instance that consists of a core process and several pockets, which behaves like a black-box, and presents a framework based on this notion. But this work does not mention how to deal with pockets at run-time.

7 Overview of the Implementation -DartFlow

The proposed service composition framework has been applied in a system named DartFlow which is a sub-project of DartGrid. DartFlow targets towards providing a convenient and efficient way for TCM workers and organizations to collaborate with each other in research activities and experiments. DartFlow has supported the service registration and query at the semantic level based on ontology inference. So far we have established the TCM domain ontology covering about 8000 class concepts and 50,000 instance concepts [7]. At present, we extend the workflow specification XPDL [14] to support the black-box element and use the extended workflow engine to support the execution of service compositions. In order to ensure the process specification valid, we transfer the extended XPDL into Petri-net, upon which we carry out verification for the process. In the test bed for the prototype in the TCM analysis scenario, we have established more than 300 rules. Due to the limitation, we do not introduce the details of the implementation. For the details, please refer to our pre-published papers before [12, 15, 16].

8 Conclusion and Future Work

This paper proposes a framework for developing and managing service composition using a flexible workflow method, in which the black-box mechanism is used to improve the flexibility of service composition. And also, it introduces the implementation of this framework in DartFlow- a service composition system briefly. The framework has the following characteristics. (1) It utilizes the black-boxes to reduce the complexity of service compositions and alleviate the workload for service composition designers. (2) It realizes the automatic service composition in part based on rules at run-time. (3) It has the ability to deal with those service compositions which can not be predefined completely. (4) It provides great flexibility to adapt to frequent changes arising from domain/business/user rules and the dynamic Internet environment. At present, the implementation of the framework in DartFlow does not consider the conflicts in rules when generating service compositions for black-boxes automatically at run-time. In the future, we will propose a mechanism to inspect the conflicts in rules and perfect the algorithms of concretizing black-boxes after considering the conflicts.

Acknowledgement

This work is supported by China 973 fundamental research and development project: The research on application of semantic grid on the sharing of knowledge and service of Traditional Chinese Medicine; and Zhejiang Provincial Natural Science Foundation of China (NO. Y105463).

References

1. F. Casati, M. C. Shan, Dynamic and adaptive composition of e-services, Information system, 26(3):143-162, 2001.
2. B. Benatallah, M.Dumas, Q. Z. Sheng, The SELF-SERV Environment for Web Services Composition, IEEE Internet Computing, 7 (1), 40-48, 2003.

3. K. Sivashanmugam, J. Miller, A. Sheth, K. Verma, Framework for Semantic Web Process Composition, Semantic Web Services and Their Role in Enterprise Application Integration and E-Commerce, International Journal of Electronic Commerce, 9(2):71-106, 2004.
4. http://www.activebpel.org/
5. J. Peer, Web Service Composition as AI Planning – a Survey, http://elektra.mcm.unisg.ch/pbwsc /docs/pfwsc.pdf, 2005.
6. S. Dustdar, W. Schreiner, A survey on web services composition, International Journal of Web and Grid Services, 1(1):1-30, 2005.
7. X. Zhou, Z. Wu, Ontology Development for Unified Traditional Chinese Medical Language System. Journal of Artificial Intelligence in Medicine, 32(1):15-27, 2004.
8. Z.H. Wu, H.J Chen, S.G. Deng, Y. Mao, DartGrid: RDF-Mediated Database Integration and Process Coordination Using Grid as the Platform, In: Proceeding of the 7th Asia-Pacific Web Conference on Web Technologies Research and Development, ApWeb, 2005.
9. Z.H. Wu, S.M Tang, S.G Tang, 2005. DartGrid II: A Semantic Grid Platform for ITS. IEEE Intelligent Systems 20(3):12-15, 2005.
10. W.M.P. van der Aalst, M. Weskez, Advanced Topics in Workflow Management: Issues, Requirements, and Solutions, Journal of Integrated Design and Process Science, 7(3):49-77, 2003.
11. I. Wetzel, R. Klischewski, Serviceflow Beyond Workflow? Concepts and Architectures for Supporting Inter-Organizational Service Processes, In: Proceeding of the 14th International Conference Advanced Information Systems Engineering, CAiSE, 2002.
12. S.G. Deng, Z.H. Wu, Management of Serviceflow in a Flexible Way, In: Proceeding of the 5th International Conference on Web Information Systems Engineering, WISE, 2004.
13. B. Esfandiari, V. Tosic, Towards a Web service composition management framework, In: Proceeding of the IEEE International Conference on Web Services, ICWS, 2005.
14. http://www.wfmc.org/standards/XPDL.htm
15. Z.H. Wu, S.G. Deng, Y. Li, Introducing EAI and Service Components into Process Management. In: Proceeding of the IEEE International Conference on Services Computing, SCC, 2004.
16. L. Kuang, J. Wu, S.G. Deng, Y. Li, Exploring Semantic Technologies in Service Matchmaking, In: Proceeding of the 3th IEEE European Conference on Web Services, ECOWS, 2005.
17. P. albert, L. Henocque, M. Kleiner, Configuration Based Workflow Composition, In: Proceeding of the IEEE International Conference on Web Services, ICWS, 2005.
18. W. Blanchet, E. Stroulia, R. Elio, Supporting Adaptive Web-Service Orchestration with an Agent Conversation Framework, In: Proceeding of the IEEE International Conference on Web Services, ICWS, 2005.
19. P. Alvarez, J.A. Ba˜nares, J. Ezpeleta, Approaching Web Service Coordination and Composition by Means of Petri Nets. The Case of the Nets-within-Nets Paradigm, In: Proceedings of Third International Conference on Service-Oriented Computing, ICSOC 2005.
20. S. Narayanan, S. McIlraith, Simulation, verification and automated composition of Web service. In: Proceedings of the 11th International World Wide Web Conference, WWW, 2002.
21. S. Sadiq, W. Sadiq, M. Orlowska, A Framework for Constraint Specification and Validation in Flexible Workflows. Information Systems, 30(5): 349-378. 2005.

Mediation Enabled Semantic Web Services Usage[⋆]

Emilia Cimpian, Adrian Mocan, and Michael Stollberg

Digital Enterprise Research Institute,
Institute for Computer Science, University of Innsbruck,
Technikerstrasse 21a, A-6020 Innsbruck, Austria
firstname.lastname@deri.org

Abstract. The Semantic Web services has become a challenging research topic in the last half of decade. Various frameworks offer means to semantically describe all the related aspects of Semantic Web services, but the solutions to the heterogeneity problems, inherent in a distributed environment such as the Web, are still to be properly integrated and referred to from the main phases of the Web services usage. Both data and process heterogeneity, as well as the multitude of functionalities required and offered by semantic Web services' requesters and providers hamper the usability of Web services, making this technology difficult to use. This paper emphasizes the role of mediators in a Semantic Web services architecture, illustrating how the mediators can enable the Semantic Web services usages in operations like discovery, invocation and composition.

1 Introduction

An intense research activity regarding Semantic Web, Web services and their combination, Semantic Web services, has been going on during the last years. But only the semantic descriptions attached to data or to the Web services deployed using today's technologies, does not solve the heterogeneity problem that may come up due to the distributed nature of the Web itself. As such, the heterogeneity existing in representing data, in the multitude of choices in representing the requested and the provided functionalities, and in the differences in the communication patterns (public processes) are problems that have to be solved before being able to fully benefit of the semantic enabled Web and Web services. Considering that these problems can not be avoided, dynamic mediation solutions that fully exploit the semantic descriptions of data and services are required.

This paper emphasizes the importance of the mediators for the usage of Semantic Web services, showing why the basic phases needed for Semantic Web services usage (discovery, invocation and composition) can hardly take place without the support of mediators. It also identifies different levels of mediation, illustrating what type of mediation is needed in a particular phase.

[⋆] The work is funded by the European Commission under the projects ASG, DIP, enIRaF, InfraWebs, Knowledge Web, Musing, Salero, SEKT, Seemp, SemanticGOV, Super, SWING and TripCom; by Science Foundation Ireland under the DERI-Lı́on Grant No.SFI/02/CE1/I13; by the FFG (Österreichische Forschungsförderungsgesellschaft mbH) under the projects Grisino, RW^2, SemNetMan, SeNSE, TSC, OnTourism.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 459–473, 2006.
© Springer-Verlag Berlin Heidelberg 2006

The discussion is held in the context of Web Service Modeling Ontology (WSMO) [4], a framework that offers all the necessary instruments to semantically describe the Web services and all the related aspects. One of the main reasons in choosing WSMO as the semantic framework for Web services is that it realizes the importance of mediators and treats them as first class citizens. WSMO offers specific means to semantically describe concrete mediation solutions and to directly refer to them where needed (e.g. from ontologies or Web services).

The paper is structured as follows: Section 2 provides an overview of Semantic Web services definition, as an important aspect for the usability of the services; Section 3 describes how the discovery, invocation and composition can benefit from the use of mediators, and what type of mediation is needed in each of these phases; Section 4 presents an illustrative example, addressing all types of mediation previously identified, while Section 5 provides an overview of the related mediation work; finally, Section 6 concludes the paper.

2 Semantic Web Services Definition

Any Semantic Web service is accessible via its interface, which provides information on how a service can be invoked. As a consequence, we believe that the ability of a service to participate in complex interactions directly depends on the expressivity of its interface and on its correctness (from the business logic point of view). Simultaneously, a service has to correctly and completely advertise its functionality (that is, what the service can provide), which will enable the service's discovery by potential requestors.

Web Service Modeling Ontology (WSMO[1]) provides an exhaustive definition of Semantic Web services [4].

Table 1. WSMO Web Service Definition

```
Class webService
  hasNonFunctionalProperties type nonFunctionalProperties
  importsOntology type ontology
  usesMediator type ooMediator, wwMediator
  hasCapability type capability
    multiplicity = single-valued
  hasInterface type interface
```

The WSMO service definition consists of the following elements:

non-functional properties - general information about the Web service, like `creator`, `format`, or `description` [12];

imported ontologies - external ontologies used in defining the service;

used mediators - different mediators needed for definig the service (for example, for importing ontologies);

capability - a functional description of what the service can do;

interface - the way of communicating with the requestor or with other services.

[1] See `http://www.wsmo.org`

From the service's behaviour point of view, the important items from this definition are the capability and the interface, and we reproduce their definitions from [12].

Table 2. WSMO Web Service Capability

```
Class capability
  hasNonFunctionalProperties type nonFunctionalProperties
  importsOntology type ontology
  usesMediator type ooMediator, wgMediator
  hasSharedVariables type sharedVariables
  hasPrecondition type axiom
  hasAssumption type axiom
  hasPostcondition type axiom
  hasEffect type axiom
```

Apart from the non-functional properties, the imported ontologies and the used mediators, the capability definition of a semantic Web service must contain the following information:

precondition - the information space of the Web service before its execution;
assumption - the state of the world before the execution of the Web service;
postcondition - the information space of the Web service after its execution;
effect - the state of the world after the execution of the Web service;
shared variables - variables that are shared between preconditions, postconditions, assumptions and effects.

Table 3. WSMO Web Service Interface

```
Class interface
  hasNonFunctionalProperties type nonFunctionalProperties
  importsOntology type ontology
  usesMediator type ooMediator
  hasChoreography type choreography
  hasOrchestration type orchestration
```

The semantic Web service's interface must contain information about the choreography and the orchestration of a service. The choreography offers indications about how a client should invoke the service, while the orchestration shows how the service can communicate with other services in order to achieve a common functionality.

3 Semantic Web Services Usability

The previous section provided some details on how WSMO currently defines Semantic Web services. An important aspect that needs to be noted is the presence of the `usesMediator` attribute in the presented definitions, showing that in WSMO, mediation can be supported by all constituent elements of a WSMO service description.

This section elaborates on *why* and *how* different types of mediation should be used for the actual usage of a service, during discovery, invocation and composition phases.

3.1 Web Service Discovery

The Web service discovery has the role of determining appropriate Web services for fulfilling a certain goal, out of a collection of services. There are numerous techniques for Web service discovery; WSMO addresses three possible discovery techniques: keyword-based discovery, discovery based on simple semantic descriptions of services, and discovery based on rich semantic descriptions of services [7].

While the keyword-based discovery can take place without the use of any mediation service, the last two (called semantic-based discovery techniques) can benefit from `data level mediation` and `functional level mediation`.

In this context the data mediation is considered to be the mediation between two ontologies. That is, the data mediator is able to transform instances expressed in terms of one ontology (considered to be the `source` ontology) in instances expressed in terms of the other ontology (`target` ontology) [10].

During the discovery, the data level mediation is needed in case the requestor uses a different ontology than the one used by the available Web services. For example, if a service's declared functionality is to provide accommodation in a certain city in Austria and train tickets between any location in Europe and that particular city, it may have in its internal ontology a concept called `train_ticket`. On the other hand, a requestor of a train ticket may have an equivalent concept called `travel_voucher`. Without the services of a data mediator able to map the `train_ticket` concept to the `travel_voucher` concept, the service could not be discovered. Of course, this is a very simple example of data mediation. Several data mediator tools are able to solve more complex mappings. For example the data mediator tool developed as part of the Web Service Execution Environment (WSMX) takes into consideration both syntactical and structural aspects, mapping concepts based on their names (syntactical aspect), attributes and relations they are involved in (structural aspects) [10].

The functional level mediation is used to state the logical relations between the service offer and the service request. Considering a request G[2] and an offer WS, there are five types of possible relations between their functionalities [16]:

1. **equal** - meaning that WS offers exactly the functionality required by G.
2. **plug-in** - meaning that WS provides all the functionalities required by G, and some extra functionalities (G is a plug-in of WS). In the previously described example, the relation between the functionality requested by the goal (booking a train ticket) and the one offered by the Web service (booking both a train ticket and an accommodation) is a plug-in relation.
3. **subsume** - G requires more functionalities that WS provides (G subsumes WS). An example for this case is if a requester asks for a complete holiday package (flight tickets and accommodation) and the service offers only accommodation.
4. **intersecting** - WS offers part of what G requests, and some additional functionality as well. An example for this case would be when a requester asks for flight tickets and accommodation and the service offers accommodation and car rental.
5. **disjoint** - the requested and provided functionalities are totally different.

[2] In WSMO the requests are expressed as `Goals`; a goal has a similar definition as a Web service, expressing in a formal way both the requested capability and the requested interface.

For a Web service to be discovered as a candidate in fulfilling a certain goal, the relation between their capabilities has to be `equal` or `plug-in`.

We might assume that these functional relationships between the requested capability and the offered capability can be derived directly in the discovery process in an automatic manner (using reasoning techniques). There also could be cases when the human domain expert has to be involved and semi-automatically (or even manually) derive these functional relations. Such situations appear when there are dependencies between the matching functionality and the remaining, additional functionality, or when financial conditions have to be analyzed.

3.2 Web Services Invocation

After the discovery of a service able to fulfil a certain request, the actual invocation can take place. Since both the provider and the requester of a service express the way they want to communicate by using the interface (choreography) description prior to the discovery, it is quite possible that there are a number of mismatches between these descriptions.

Some of them can be solved by data mediation techniques (like the `train_ticket - travel_voucher` one illustrated in the previous section), but some of them can be communication specific, solvable only by using `process level mediation` techniques (we call this process mediation since a choreography represents the public processes of an entity).

[2] identifies five types of mismatches that can be automatically solved, considering that the choreographies are expressed conforming to the Abstract State Machine (ASM) specifications [1][3], as illustrated in the following figure.

Stopping an unexpected message (Figure 1. a)): in case one of the partners sends a message that the other one does not want to receive, the mediator should just retain and store it. This message can be send later, if needed, or it will just be deleted after the communication ends.

Inversing the order of messages (Figure 1. b)): in case one of the partners sends the messages in a different order than the one the other partner wants to receive them. The messages that are not yet expected will be stored and sent when needed.

Splitting a message (Figure 1. c)): in case one of the partners sends in a single message multiple information that the other one expects to receive in different messages.

Combining messages (Figure 1. d)): in case one of the partners expects a single message, containing information sent by the other one in multiple messages.

Sending a dummy acknowledgement (Figure 1. e)): in case one of the partners expects an acknowledgement for a certain message, and the other partner does not intend to send it, even if it receives the message.

3.3 Web Services Composition

The Web service composition is the most complex action from the three phases described in this paper, and involves the previously two described actions.

[3] WSMO also uses ASM for representing the choreographies [15].

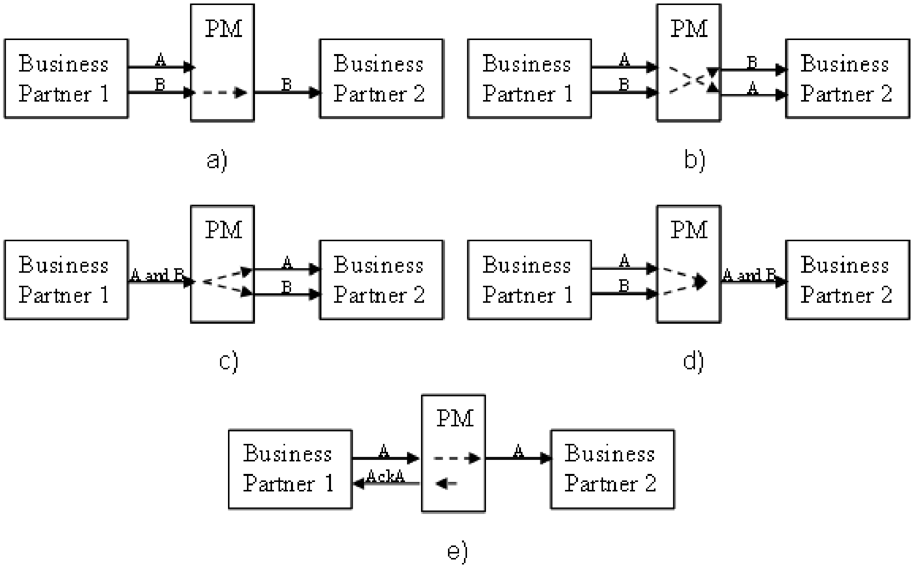


Fig. 1. Solvable Mismatches

Composition and Discovery - discovery is needed for composition in order to identify the services that need to be composed; any service that offers only part of the required functionality is a candidate for the composition.

Composition and Invocation - the composition of several services can be seen as a composition of several invocations; in order to compose different services, an execution environment needs to be able to communicate with all of them, sequentially or in parallel.

Similarly with the relation that should exist between a goal and a Web service (for the Web service to be discovered as a candidate for fulfilling the goal) the relation between the goal and the composition of services has to be `equal` or `plug-in`. In any other situation the composition is not correct or not complete (for example, if the required functionality subsumes the functionality offered by the composition of Web services, the composition is not complete - one or more other services need to be added to the composition). In the following subsections, we will analyze the possible relations between the goal and the composed Web services, that would allow obtaining a valid composition.

For determining these relations between the functionality of the goal and the functionality of the potential services and the composed functionality, the functional mediation can be used. Also the data level mediation may be needed in case different ontologies are used for representing data.

The help of a process mediator is needed during the actual invocation of the composed services, if the behavior of the participants differ.

Exact Match. The exact match between the functionality offered by the composition of the services and the one requested by the goal can appear only if

- all the composed services offere functionalities subsumed by the required functionality,

and

- the requested functionality is the exact reunion of the functionalities offered by the services.

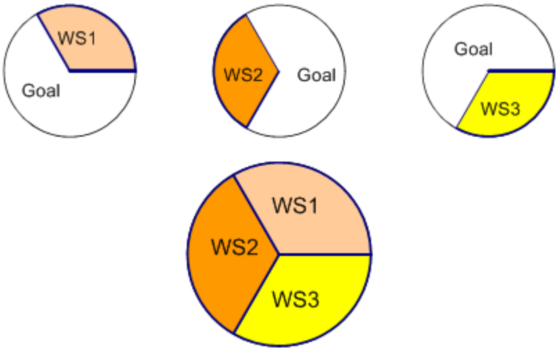


Fig. 2. Exact Match

Please note that the exact match between the goal's functionality and the functionality of the composed services still stands even if the functionalities of individual services overlap at some point.

An example of such a mach is the following:

G - requests train tickets between two locations in Austria, hotel reservation in the destination city and car hiring in the destination city;
WS_1 - offers train tickets between any two locations in Austria;
WS_2 - offers hotel accommodation in any city in Austria;
WS_3 - offers cars for rent in any city in Austria.

Plug-in match. The plug-in match between the functionality required by the requestor and the functionalities offered by the composition of the services (i.e. composition subsumes the requested functionality) can appear only if

- all the composed services offere functionalities subsumed by the required functionality, or the intersection between the required functionality and the one offered by the services is not null,

and

- the requested functionality is a plug-in of the reunion of the functionalities offered by the services.

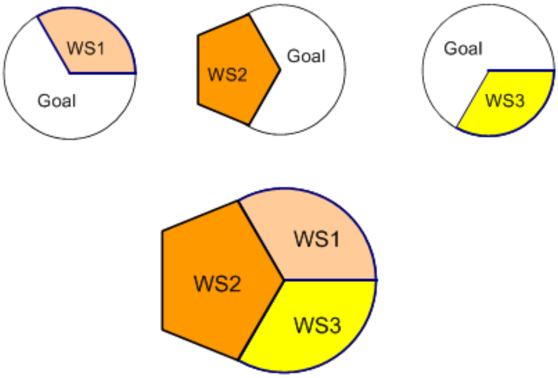


Fig. 3. Plug-in Match

Similarly with the previous case, the relation stands even if the intersection between the functionalities of individual services overlap at some point.

An example of such a mach is the following:

G - requests train tickets between two locations in Austria, hotel reservation in the destination city and car hiring in the destination city;
WS_1 - offers train tickets between any two locations in Austria;
WS_2 - offers hotel accommodation in any city in Europe (not only in Austria);
WS_3 - offers cars for rent in any city in Austria.

4 Example

In order to illustrate how the mediation can be used in various stages of services' usages, we consider the following example: a client requesting train tickets between two Austrian cities, and hotel accommodation in the destination city. The available services are offering: WS_1- train tickets between any two cities in Europe, WS_2 - hotel accommodation in any city in Austria.

The lack of space does not allow us to give all the details regarding the goal, Web services, ontologies, and mismatches that may appear in such a scenario. The following sections illustrate some possible data and process mismatches, and the way these could be solved, and also the functional relations between the goal and WS_2 and the goal and the service composition [4].

4.1 Data Mediation

For illustrating the data mismatches and how the mismatches can be solved, we consider the following example.

The goal has in its ontology the concept `station` with the following definition:

[4] All the examples are expressed using Web Service Modeling Language (WSML): `http://www.wsmo.org/wsml/`

```
concept station
 nonFunctionalProperties
   dc5#description hasValue "Station concept"
 endNonFunctionalProperties
 start_Location typeOf _boolean
 destination_Location typeOf _boolean
 name typeOf _string
```

where start_Location and destination_Location are two boolean attribute showing if the station represents the starting or the ending point of a trip, and name is the actual name of a station. For example, an instance of the station concept S having the start_Location set to true, and the destination_Location to false (assuming that internally there is an imposed condition on these attributes, that only one of them can be true) will be considered to be the starting point of a trip.

On the other hand the service WS_1 may have in its ontology the concept route, with the following definition:

```
concept route
 nonFunctionalProperties
   dc#description hasValue "Route concept"
   endNonFunctionalProperties
 from typeOf _string
 to typeOf _string
```

showing which are the names of departure and arrival stations.

Without the services of a data mediator, an execution environment would not be able to determine that from two instances of station an instance of route have to be created. For supporting this, a data mediator has to be able to create and execute rules similar with the following ones[6]:

```
Mapping(
  OG7#station
  OS8#route
  classMapping( one-way station route))

Mapping(
  OG#destination_Location
  OS#to
  attributeMapping( one-way
    [(station) destination_Location => boolean]
    [(route) to => string]))
    valueCondition
      (station [(station) destination_Location => boolean] true)

Mapping(
  OG#start_Location
  OS#from
  attributeMapping( one-way
    [(station) start_Location => boolean]
```

[5] dc is the prefix we use to refer to Dublin Core non-functional properties set URL http://purl.org/dc/elements/1.1

[6] The rules are expressed in the Abstract Mapping Language [14] are generated and can be executed using the Web Service Execution Environment (WSMX) data mediation tool, available for download at http://sourceforge.net/projects/wsmt

[7] We use OG to denote the goal's ontology.

[8] We use OS to denote the service's ontology.

```
    [(route) from => string]))
    valueCondition
      (station [(station) start_Location => boolean] true)

Mapping(
  OG#name
  OS#to
  attributeMapping( one-way
    [(station) name => string]
    [(route) to => string]))

Mapping(
  OG#name
  OS#from
  attributeMapping( one-way
    [(station) name => string]
    [(route) from => string]))
```

The first rule states the relations between `station` and `route`, the following two between the boolean attributes from the station and the `to` and `from` attributes from the route. The last two rules are showing the relation between the `name` attribute from the station and the `to` and `from` attributes.

4.2 Functional Level Mediation

For illustrating the functional level mediation we will describe the capabilities of the goal, WS_1 and WS_2 and also the capability of the services' composition. For simplicity reasons, we will present only the post-conditions, which describe the information space of the Web service after its execution; additionally we consider that all the involved parties use the same terminology.

```
 goal G
  capability Gcapability
    postcondition Gpostcondition
      definedBy
        ?x[start_location hasValue ?cityS,
           destination_location hasValue ?cityD] memberOf ticket and
        ?cityS[locatedIn hasValue "Austria"] and
        ?cityD[locatedIn hasValue "Austria"] and
        ?h[locatedIn hasValue ?cityD] memberOf hotel and
        ?r[client hasValue ?p,
           hotel hasValue ?h] memberOf Reservation and
        ?p memberOf person.[9]

 webService ws1
  capability ws1capability
    postcondition ws1postcondition
      definedBy
        ?x[start_location hasValue ?cityS,
           destination_location hasValue ?cityD] memberOf ticket and
        ?cityS[locatedIn hasValue "Europe"] and
        ?cityD[locatedIn hasValue "Europe"].

 webService ws2
  capability ws2capability
    postcondition ws2postcondition
      definedBy
        ?h[locatedIn hasValue ?city] memberOf hotel and
```

[9] ”?” denotes a variable in WSML.

```
        ?city[locatedIn hasValue "Austria"] and
        ?r[client hasValue ?p,
           hotel hasValue ?h] memberOf Reservation and
        ?p memberOf person.

 webService composedws
  capability composedwscapability
    postcondition composedwspostcondition
      definedBy
        ?x[start_location hasValue ?cityS,
           destination_location hasValue ?cityD] memberOf ticket and
        ?cityS[locatedIn hasValue "Europe"] and
        ?cityD[locatedIn hasValue "Europe"] and
        ?h[locatedIn hasValue ?city] memberOf hotel and
        ?city[locatedIn hasValue "Austria"] and
        ?r[client hasValue ?p,
           hotel hasValue ?h] memberOf Reservation and
        ?p memberOf person.
```

The functional relation between G and WS_2 has to illustrate the fact that WS_2 offers less than what the goal requests [10]:

```
source G
target WS2
nonFunctionalProperties
  dc#type hasValue subsume[11]
endNonFunctionalProperties
  definedBy
    ?h[locatedIn hasValue ?cityD] memberOf hotel and
    ?r[client hasValue ?p,
        hotel hasValue ?h]
    memberOf Reservation and
    ?p memberOf person.
```

The functional relation between the goal and the composition of services has to illustrate the fact that the composition offer more functionality then the one required by the goal.

```
source G
target composedws
nonFunctionalProperties
  dc#type hasValue plug-in
endNonFunctionalProperties
  definedBy
    ?x[start_location hasValue ?cityS,
       destination_location hasValue ?cityD] memberOf ticket and
    naf[12] ?cityS[locatedIn hasValue "Austria"] and
    naf ?cityD[locatedIn hasValue "Austria"].
```

4.3 Process Level Mediation

For illustrating process mediation mismatches we will provide a piece of the goal choreography and and a piece of WS_1 choreography. Both of them are representing using WSMO definition of choreographies [15], which respects the ASM specifications.

[10] The relations are expressed based on [16]; this is still on-going work, and there is no tool available for generating them.

[11] This non-functional-property is used to express the way the subsumption relation should be read, i.e G subsumes WS2 by ...

[12] naf stands for negation as failure in WSML [8].

In WSMO, the owner of any instance (that is, the concept that is instantiated) expected by an entity is part of an `in` list, and any owner of an instance that should be sent by an entity is part of an `out` list. Further more, the order in which the instances are expected is set by using transition rules, consisting of conditions and updates.

The goal choreography contains the following rules:

```
/*
* the invocation starts with the creation of a date instance; no condition
* need to be fulfilled in order to create this instance
*/
do
  add(_#[
    year hasValue ?year,
    month hasValue ?month,
    day hasValue ?day
  ]memberOf tro#date)

/*
* after the date is created, the requestor creates an instance of station -
* the starting point of the trip
*/
forAll ?date with (?date[] memberOf
tro#date
  ) do
    add(_#[
      start_Location hasValue _boolean("true"),
      destination_Location hasValue _boolean("false"),
      name hasValue ?name
    ]memberOf tro#station)
endForAll

/*
* after the instance denoting the starting point of the trip exists, the
* requestor creates an instance denoting the destination point
*/
forAll ?station with (?station[
  start_Location hasValue _boolean("true"),
  destination_Location hasValue _boolean("false"),
  ] memberOf tro#station
  ) do
    add(_#[
      start_Location hasValue _boolean("false"),
      destination_Location hasValue _boolean("true"),
      name hasValue ?name
    ]memberOf tro#station)
endForAll
```

where `station` and `date` are both part of the `out` list of the choreography.

The choreography of WS_1 contains the following rules:

```
/*
*the invocation starts when the service receives an instance of route
*/
do
  add(
    _#[ from hasValue ?from,
    to hasValue ?to
  ]memberOf too#route)

/*
* after the route is created, the service expects an instance of date - the
* date of the trip
*/
```

```
forAll ?route with (?route[] memberOf
too#route
  ) do
    add(_#[
      year hasValue ?year,
      month hasValue ?month,
      day hasValue ?day
    ]memberOf too#date)
endForAll
```

where route and date are part of the in list.

A graphical representation of the two choreographies is illustrated in Figure 4. a).

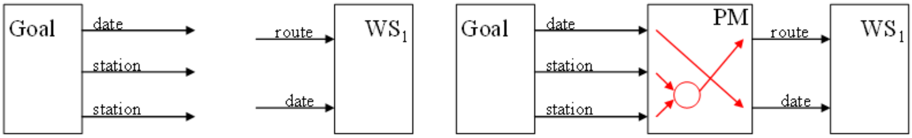


Fig. 4. Choreography Mismatches

What the process mediator should do in this case is represented in Figure 4. b). For example, the way a Process Mediator could work is [2]:

- when the instance of date is sent by the requestor, the process mediator should determine that it will be expected at some point in time (by checking the service's in list) but it is not expected at the beginning of the conversation. As a consequence, the date instance should be stored for further use.
- when the first instance of station is received, the process mediator should invoke a data mediator in order to obtain the equivalent instance in terms of the service's ontology. The data mediator will return an instance of route, which conforming with the service's choreography is expected at this point of the communication. The problem is that this instance is incomplete, it does not have all the attributes instantiated as required, so the process mediator will store the station instance for further use.
- when the second instance of station is sent, the process mediator invokes a data mediator with both station instances, obtaining a correct instance of route from the service's point of view. As the route is expected, this instance is sent to the service.
- after the route is sent, the process mediator determines, based on the service's choreography that an instance of date is expected now. The date instance is retrieved from the internal storage and sent.

The WSMX process mediator prototype[13] is able to perform this kind a computations, and to address all the types of mismatches identified in [2].

[13] Available for download from http://sourceforge.net/projects/wsmx/

5 Related Work

Data mediation represents an old research topic that was reshaped and re-explored in the semantic context. Semantic-based solution have been proposed that offer better, more-dynamic and Web oriented mediators in a more-effective and effort saving manner [9,11,3].

At the same time, processes mediation is still a poorly explored research field. The existing work represents only visions of mediator systems able to resolve in a (semi-) automatic manner the processes heterogeneity problems, without presenting sufficient details about their architectural elements. Still, these visions represent the starting points and valuable references for the future concrete implementations (see for example Contivo[14] and CrossWorlds[15]).

As far as we know the functional mediation has not been directly addressed in any other work. However similar classification and functional relationships were explored in various discovery working groups [6,13,5] as prerequisites for the discovery engines.

Even if as future development this work aims at providing complete solution for these three areas of mediation, this paper focuses on analyzing how these complementary techniques can be integrated and used in the main steps of Semantic Web services usage. We are not aware of ay similar overview and work towards a complete mediation framework for Semantic Web services.

6 Conclusions

This paper emphasizes the importance of mediators in a Semantic Web services infrastructure, illustrating why and how the mediators can be used during Semantic Web services discovery, invocation and composition. These three phases, considered to be of highly importance for the Semantic Web services usage are explained in the paper, and the appropriate mediation technics are identified. These techniques refer to data mediation (tackling the terminology and representation mismatches), process mediation (addressing the public process mismatches, i.e., communication mismatches) and functional mediation (bridging various required and offered capabilities).

The paper also presents examples of mismatches that can appear in a Semantic Web services environment, and it proposes ways of solving these heterogeneity problems.

References

1. E. Börger and R. Stärk. *Logical Foundations of Artificial Intelligence.* Springer, Berlin, Heidelberg, 1987.
2. E. Cimpian and A. Mocan. WSMX Process Mediation Based on Choreographies. In *Proceedings of the 1st International Workshop on Web Service Choreography and Orchestration for Business Process Management at the BPM 2005, Nancy, France*, 2005.

[14] `http://www.contivo.com`

[15] `http://www.sars.ws/h14/ibm-crossworlds.html`

3. J. Euzenat, D. Loup, M. Touzani, and P. Valtchev. Ontology alignment with ola. *Proc. 3rd ISWC2004 Workshop on Evaluation of Ontology-based Tools (EON), Hiroshima, Japan*, pages 59–68, 2004.
4. C. Feier, A. Polleres, R. Dumitru, J. Domingue, M. Stollberg, and D. Fensel. Towards intelligent web services: The web service modeling ontology (WSMO). *International Conference on Intelligent Computing (ICIC)*, 2005.
5. H.-C. Hsiao and C.-T. King. Neuron - A Wide-Area Service Discovery Infrastructure. *In Proceedings of the Internaltional Conference on Parallel Processing (ICPP'02)*, 2002.
6. U. Keller, R. Lara, H. Lausen, A. Polleres, and D. Fensel. Automatic Location of Services. In *Proceedings of the 2nd European Semantic Web Conference (ESWC 2005), Crete, Greece*, 2005.
7. U. Keller, R. Lara, and A. Polleres (eds.). WSMO Web Service Discovery. Deliverable D5.1, 2004. available at: http://www.wsmo.org/TR/d5/d5.1/.
8. H. Lausen, J. de Bruijn, A. Polleres, and D. Fensel. WSML - A Language Framework for Semantic Web Services. *W3C Workshop on Rule Languages for Interoperability*, April 2005.
9. A. Maedche, B. Motik, N. Silva, and R. Volz. Mafra - a mapping framework for distributed ontologies. *Proceedings of the 13th European Conference on Knowledge Engineering and Knowledge Management (EKAW)*, September 2002.
10. A. Mocan and E. Cimpian. Mapping creation using a view based approach. *1st International Workshop on Mediation in Semantic Web Services (Mediate 2005)*, December 2005.
11. N. Noy. Semantic Integration: a Survey of Ontology-based Approaches. *ACM SIGMOD Record*, 33(4):65–70, 2004.
12. D. Roman, H. Lausen, and U. Keller (eds.). Web Service Modeling Ontology (WSMO). Deliverable D2, 2005. available at: http://www.wsmo.org/TR/d2/.
13. B. Sapkota, L. Vasiliu, I. Toma, D. Roman, and C. Bussler. Peer-to-peer technology usage in web service discovery and matchmaking. *Sixth International Conference on Web Information Science and Engineering (WISE 2005)*, November 2005.
14. F. Scharffe and J. de Bruijn. A language to specify mappings between ontologies. In *Proc. of the Internet Based Systems IEEE Conference (SITIS05)*, 2005.
15. J. Scicluna, A. Polleres, and D. Roman (eds.). Ontology-based Choreography and Orchestration of WSMO Services. Deliverable D14, 2005. available at: http://www.wsmo.org/TR/d14/.
16. M. Stollberg, E. Cimpian, and D. Fensel. Mediating Capabilities with Delta-Relations. In *Proceedings of the First International Workshop on Mediation in Semantic Web Services, Amsterdam, the Netherlands*, 2005.

Toward Automatic Discovery and Invocation of Information-Providing Web Services*

Wen-feng Zhao and Jun-liang Chen

State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications (BUPT), Beijing, China
zhaowenfeng@gmail.com, chjl@bupt.edu.cn

Abstract. Semantic Web makes the automatic discovery and invocation of Web Services become possible. But existing methods perform the capability matching, which is crucial for service discovery, either only according to inputs and outputs (IO), which results in a not very precise matching, or trying to tackle arbitrary services, which results in an undecidable reasoning. In this paper, targeting merely the information-providing type of Web Services, we present a precise and decidable matching method based on the Description Logic reasoner. An outstanding property of our method is that it can determinate the accurate binding of IO between requested and advertised services, which is necessary for automatic invocation yet rarely addressed in previous work. Besides, this paper also describes a useful use case for automatic Web Services discovery and invocation.

1 Introduction

Semantic Web Services, i.e. Web Services with Semantic markup, are widely considered capable of providing a computing environment where different machines are able to not only interoperate in syntax but also understand in semantics each other, which will make possible the automation of a variety of Web Services related tasks such as discovery, selection, invocation, composition and execution monitoring. Among these tasks, the automatic discovery and invocation are in a more basic place and probably will be realized prior to others, since automatic discovery can be seen as a special case of automatic composition which has only one component in the composition result.

Such a perspective is approaching in recent years with the emergence of semantic markup languages for Web Services such as OWL-S, WSDL-S and WSMO. These recommendations have a very similar mechanism in description of services' capability, namely mainly through semantic annotation to such properties of a service as inputs, outputs, preconditions, effects (IOPE), and categories.

Utilizing them, the automatic Web Services discovery has been studied quite a bit. However, most of such research exploits only an IO-based capability matching method sometimes in conjunction with a simple category-based one [7, 4, 8, 10].

* This work is supported by the National Natural Science Foundation of China under the project "Fundamental Research for Intelligent Mobile Service Platform" (Grant No: 60432010).

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 474 – 480, 2006.
© Springer-Verlag Berlin Heidelberg 2006

According to this method, an advertised service matches a requested service in condition of every output of the request has a counterpart in the outputs of the advertisement according to their associated ontology concepts, and every input of the advertisement has its counterpart in those of the request.

Although useful in some certain cases, such IO-based method without taking PE into consideration isn't adequate to express the capability of Web Services exactly and then couldn't match services precisely. In our opinion, the bypassing around PE is because for arbitrary Web Services the PE involves arbitrary proposition axioms and hence is hard to express and reason about. First-order logic should be competent to express PE of many Web Services, but the reasoning on it is undecidable.

On the other hand, it is widely considered that Web Services fall into two types: information-providing ones and world-altering ones [5]. The languages presently assumed to express PE such as RuleML, DRS, SWRL, and OCL [5, 1] either need to be integrated with ontology languages having been adopted in Semantic Web or become undecidable after the integration [6]. In this paper, we focus only to the information-providing services instead and draw out a precise expression for PE and a decidable matching method on it.

We express PE in OWL-DL, perform the matchmaking mainly on IOPE, and when matching occurs present the best IO binding between requested and advertised services even for services with more than one input or output relating to same ontology concept. Obviously the determination of IO binding is absolutely necessary when the matched service needs to be invoked automatically, but is usually ignored in previous research.

2 A Motivating Scenario

To illustrate the potential value of automatic discovery and invocation of Web Services, let us examine an interesting scenario called comparison shopping, i.e. listing a variety of quoted prices for a specified product from different on-line shops together for pre-shopping decision. Obviously, if implemented efficiently, such an application will be very attractive for customers since it can take rather full advantage of the web and, especially, bring those valuable small web sites to customers.

Today's comparison shopping web sites collect the quotation from original shopping sites either by "screen scraping" which parses text intended for human viewers or by being fed with products data from shopping sites in the format specified by the comparison sites. Either of them needs the interface between each comparison site and each shopping site to be negotiated manually and respectively. As a result, a comparison site could only cover some but not most of shopping sites.[1]

When employing Semantic Web Services, a more complete price comparison could be presented. Suppose in future most shopping sites provide Web Services with semantic markup to expose their product information, then we can realize the comparison site through the automatic discovery and invocation of Web Services (see Figure 1). The new comparison site will employ certain service templates, i.e. Web

[1] We've verified this by retrieving a specified book, *Gone with the Wind* with ISBN 0446365386, at several popular comparison sites such as Shopping.com, BizRate, PriceGrabber and BookFinder.

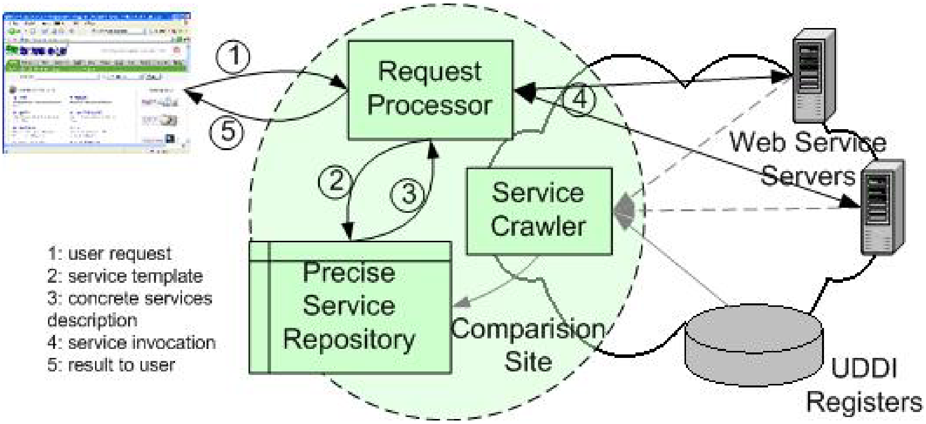


Fig. 1. Semantic Web Services Based Comparison Shopping

Services interface without implementation. The inputs of the templates are bound to input fields in the web page shown to end users and the outputs are bound to certain result fields to be listed to users.

In background a *Service Crawler* timely collects the services matching the capability requirement of the service templates from Web Services registries (of UDDI or other types) and maybe shopping sites directly (by crawling OWL-S/WSDL-S/WSML files on them) across Internet. The descriptions of matched services are stored in *Precise Service Repository*. During runtime, after a user submits a merchandise ID (e.g. ISBN for a book) to request quotations, *Request Processor* fetches all the concrete Web Services matching the corresponding template directly from the *Precise Service Repository*, and invokes them respectively. At last, *Request Processor* lists the returned data from diverse shopping sites together to the user through web pages. All these steps are performed automatically.

To put such a system into reality, a key challenge is just the precise Web Services capability matching that enables the discovered services to be invoked without human intervention. We address it below.

3 IOPE-Based Web Services Capability Matching Method[2]

3.1 Representation of PE

PE of information-providing Web Services could be expressed by axioms of Description Logic such as OWL-DL in our opinion. It is because the function of such services involves no real world states but only relations between outputs and inputs. Such case is just like that of traditional relational database where a query is expressed through relations between different entities.

[2] The concept "Web Service" in this paper means the single operation Web Service, or in fact one operation of a general Web Service, as regarded in similar research [7, 8].

The concrete expression is straightforward. We use the *Class* (i.e. *Concept* in some DL) from OWL to annotate the semantic of IO as previous work [7], and use *ObjectProperty* with IO variable names as parameters to annotate the binary relations between IO variables in PE. General n-ary (n>2) relations could be expressed by introducing into ontology special *Class* each property of which represents a dimension of the relation. And like the fields that joint different tables but don't appear in results in SQL query statements, a temporary type of variables is introduced. In this paper, we call such variable type as *Local* (L). This notion corresponds to the elements *Local* and *ResultVar* in OWL-S.

For example, suppose there is an ontology in food industry which contains a Class *Wine* with an ObjectProperty *madeIn*, and a Class *Sale* with at least two ObjectProperties *sellingProduct* and *sellingArea*, then the two "region to wines" services mentioned in [5] could be depicted respectively with "Effect: **madeIn(out, in)**" and "Local: **Sale(x)** and Effect: **sellingProduct (x, out)⊓sellingArea(x, in)**", given "in/out" are the names of input/output of the two services. The empty precondition means a logic expression with a constant value, namely *true*.

It's worthy to point out that such PE usage isn't very consistent with OWL-S where PE are used to express the real world states and usually irrelevant with information-providing services [5]. We believe the generalization like here is necessary in order to express the capability of information-providing services. In fact, in WSMO, the information space constraints and real world states are all supported respectively by *Precondition/Postcondition* and *Assumption/Effect.*

Presently we only consider the axiom with form of conjunction of atom axioms as in the above example. The conditional form of effect in OWL-S (i.e. *Result*) can be tackled by representing a single OWL-S service through several virtual inner services, which isn't detailed here for space reason.

3.2 Matching Rationale

There are different types of capability matching for Web Services [11]. As to the requirement of automated discovery and invocation, the assertion that a service S is capable of matching service R means that R could be replaced by S without changing the function it provided to its user.

The capability of a Web Service could be formally represented as an implication axiom: P→E. When regarding the domain knowledge base (KB) as the axiom set Ã, we can formalize that R can be replaced by S as:

$$\Gamma \models (P_S \to E_S) \to (P_R \to E_R) \tag{1}$$

It's too complex to be checked by DL reasoner. We consider one of its sufficient conditions:

$$\Gamma \models (P_R \to P_S) \wedge (E_S \to E_R) \tag{2}$$

Axiom (2) equals to the following two axioms being satisfied simultaneously:

$$\boldsymbol{\Gamma, P_R \models P_S} \text{ , and } \boldsymbol{\Gamma, E_S \models E_R} \tag{3}$$

It indicates that we can check if S could replace R through the following steps:

Step 1. Add clauses of E_S into the KB, check by DL reasoner whether enough many clauses of E_R could be satisfied, then withdraw the newly added clauses;
Step 2. Add clauses of P_R into the KB, check by DL reasoner whether enough many clauses of P_S could be satisfied, then withdraw the newly added clauses;

If the two conditions are all satisfied, we can say R can be replaced by S, i.e. S matches R definitely in capability. What should be noticed is that (2) is only a sufficient but not necessary condition, although it's a very popular and strong one.

3.3 Concrete Matching Steps

In detail, before executing the two steps in section 3.2, we have to set up the binding of IOL between R and S because these variables are always assigned to different names in different services. Besides, the matching on IO itself also demands the IO binding so that every output of R could be satisfied by an output of S and every input of S could be satisfied by R.

As a result, two sets of injections, **{F_O: $O_R \rightarrow O_S$}** and **{F_I: $I_S \rightarrow I_R$}** need to be found out at first. Every pair of variables in the injections should meet a certain matching degree depicted in [7]. Such an injection set will contain more than one element when more than one variable in outputs (or inputs) is associated with same concept or two concepts with enough small semantic distance. Such variables should be distinguished because they usually behave differently in PE.

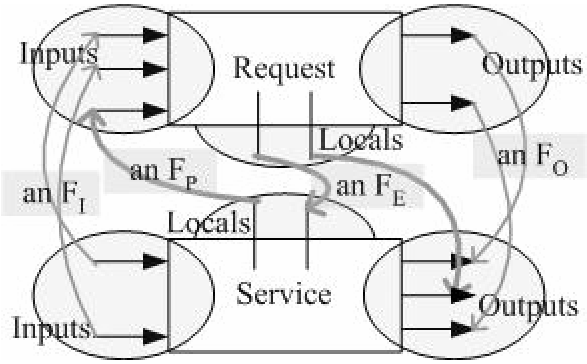


Fig. 2. An Example of F_I-F_O-F_E-F_P Combination

Next, for every combination of F_I and F_O, i.e. for every element in $\{F_I\}*\{F_O\}$, the other variables appeared in E_R and P_S need also to be matched. Therefore, for each of such combinations, the two sets of following injections need to be found out (see Figure 2):

F_E: {x | x∈$L_R \cup I_R$, x appears in E_R, and x doesn't appear in current F_I and F_O} → {y | y∈$L_S \cup 0_S$, and y doesn't appear in current F_I and F_O}

F_P: {x | x∈$L_S \cup 0_S$, x appears in P_S, and x doesn't appear in current F_I, F_O and F_E} → {y | y∈$L_R \cup I_R$, and y doesn't appear in current F_I, F_O, and F_E}

Thus, to execute step 1, for each F_I-F_O combination, we need for each F_E substitute all variables appeared in E_R with their counterparts in S, get E_R’, then check how many clauses of E_R’ can be satisfied after adding all clauses of E_S into KB. Furthermore, while executing step 2 we need find out all related F_P for each F_E,, and substitute variables in P_S according to current F_I, F_O, F_E, and F_P to get F_P’.

Now we get possibly many satisfaction degrees about Preconditions, notated as d_P, each for an F_I-F_O-F_E-F_P combination, many d_E about Effect each for an F_I-F_O-F_E combination, as well as many d_I each for an F_I and many d_O each for an F_O. By synthesizing such 4 degrees, we can get an overall matching degree d for each F_I-F_O-F_E-F_P combination. The best one of all d is just the result capability matching degree to R by S, and the related F_I and F_O form the best IO binding between R and S. To compare and/or calculate out such degrees as d, d_I, d_O, d_P and d_E, certain quantification is usually required, which is omitted in this paper for space reason. The top level algorithm is shown in figure 3.

```
iopeMatch(_request, _service, _threshold){
  find out {_fi};
  find out {_fo};
  _MapsIO = {_fi} * {_fo};
  for(each _map in _MapsIO) {
    find out {_fe} under _map;
    for(each _fe) {
      _de[_fe] = effect matching degree;
      find {_fp} under _fe;
      for(each _fp) {
        _dp[_fe, _fp] = precond matching degree;
      }
    }
    _d[_map] = Max(aggregate(_di, _do, _dp, _de));
  }
  _dMax = Max(_d[_map]);
  return (_dMax and its corresponding _map);
}
```

Fig. 3. Overview of the IOPE-Based Matching Method

4 Implementation, Related Work, and Discussion

We have implemented above IOPE-based matching algorithm initially in Java as a standalone none-GUI system, called WS CapMatcher. KAON2 reasoner is employed in it to perform the underlying DL reasoning tasks. To parse OWL-S files, CMU OWL-S API 1.1 and Mindswap OWL-S API 1.1.0 beta have both been tried.

Among the related IOPE-based work, [11] discussed early the capability matching of traditional software component and presented the IOPE-based approach besides others. It clarified the capability matching into several types, of which the plug-in one is just the type we discuss here. Whereas the implementation there employed undecidable multi-sorted first-order logic to express PE, which made the reasoning can not finish without manual intervention. [3, 2] express PE through a certain Horn Clause Logic with the implication operator replaced with “è-subsumption”. Its reasoning is also decidable at the cost of losing some expression power different with ours.

In general, within the narrowed domain, our matching method is both precise which enables the following automatic invocation and decidable which is ensured by the decidability of DL reasoning. Yet it performs poor at present for the injection-finding process is recursively realized and time-costing. So some certain pre-filtering mechanisms are needed, which can be simply through category property of service, or by indexing services in ontology through corresponding concepts of IO [9]. Furthermore, *DataProperty* with other issues is also required in PE expression to describe the value constraints on IO and other aspects. These are all our future work.

References

1. R. Akkiraju, J. Farrell, J. Miller, et al. Web Service Semantics - WSDL-S, Version 1.0, A joint UGA-IBM Technical Note, http://lsdis.cs.uga.edu/library/download/WSDL-S-V1.pdf, April 2005.
2. X. Gao, J. Yang, M.P. Papazoglou. The capability matching of web services. In *Proceedings of IEEE Fourth International Symposium on Multimedia Software Engineering*, pages 56-63, 2002.
3. T. Kawamura, D.J. Blasio, T. Hasegawa, et al. Preliminary report of public experiment of semantic service matchmaker with uddi business registry. In *Proceedings of 1st International Conference on Service Oriented Computing (ICSOC)*, Trento, Italy, 2003
4. L. Li, and I. Horrocks. A software framework for matchmaking based on semantic web technology[J]. *International Journal of Electronic Commerce*, 8(4): 39-60, 2004
5. D. Martin, M. Paolucci, S. McIlraith, et al. Bringing Semantics to Web Services: The OWL-S approach. In *First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, San Diego, CA, USA, 2004.
6. B. Motik, U. Sattler and R. Studer. Query Answering for OWL-DL with Rules. In *Proceeding of the 3rd International Semantic Web Conference (ISWC 2004)*, pages 549-563, 2004.
7. M. Paolucci, T. Kawamura, R.T. Payne, et al. Semantic Matching of Web Services Capabilities. In *Proceedings of First International Semantic Web Conference (ISWC 2002)*, pages 333-347, Sardinia, Italy, 2002.
8. E. Sirin, J. Hendler, and B. Parsia. Semi-automatic Composition of Web Services using Semantic Descriptions. Presented at *Web Services: Modeling, Architecture and Infrastructure* workshop at *the 5th International Conference on Enterprise Information Systems (ICEIS 2003)*, April 2003
9. N. Srinivasan, M. Paolucci , and K. Sycara. Adding OWL-S to UDDI: implementation and throughput. In *Proceeding of first International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, San Diego, CA, USA 2004.
10. K. Verma, K. Sivashanmugam, A. Sheth, et al. METEOR–S WSDI: A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services. In *Journal of Information Technology and Management*, 6(1):17-39, January 2005.
11. M.A. Zaremski, M.J. Wing. Specification matching of software components. In *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 6(4):333-369, 1997

Automatic Composition of Semantic Web Services - A Theorem Proof Approach*

Li Ye[1] and Junliang Chen[2]

[1] Beijing University of Posts and Telecommunications, Beijing 100876, China
sagi.ye@gmail.com
[2] Beijing University of Posts and Telecommunications, Beijing 100876, China
chjl@bupt.edu.cn

Abstract. This paper proposes a method to automatically generate composite services. The function of a service is specified by its Inputs, Output, Preconditions, and Result (IOPR). These functional descriptions are translated into a first-order logic (FOL) formula. An Automatic Theorem Prover (ATP) is used to generate a proof from known services (as axioms) to the composite service (as an object formula). Based on the deductive program synthesis theory, the implementation of the composite service is extracted from the proof. The "proof to program" method used here guarantees the completeness and correctness of the result. An brief introduction of the prototype system is given.

Keywords: Semantic Web Services, Automatic Service Composition, Automatic Theorem Proof, Deductive Program Synthesis.

1 Introduction

1.1 Evolutions of the WWW

The World Wide Web (WWW), invented in the late 1980s by Tim Berners-Lee, was originally composed of inter-linked web pages distributed on the Internet. Till recent years, the WWW begins an evolution to the Semantic Web [1]. By using formal description methods, such as RDF [2] and OWL [3] ontologies, Machine-Processable web contents are taking the place of Human-Readable web contents. This facilitates the application of various intelligent techniques.

Besides the semantic trends, the WWW is also undergoing another evolution: from an information repository to a service provider. Web Services, a way to Service Oriented Architecture (SOA), by taking the advantages of XML and SOAP [4] technologies, expands the component inter-operate from intra-domain to WWW-domain.

Semantic Web Services (SWS) are the result of the combination of these two evolutions. By SWS, we mean Web Services with a formal, unambiguous and Machine-Processable description of its properties, interfaces, and capabilities.

* This work was part of the project "Basic Research on Intelligent Mobile Service Platform", sponsored by the National Natural Science Foundation of China (No. 60432010).

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 481–487, 2006.
© Springer-Verlag Berlin Heidelberg 2006

1.2 Automatic Service Composition

Both the two trends of evolution share a common motive: to make things more automatic. Specially for Web Services, the mechanisms to automate the tasks of discovering, invoking, composing, and verifying services are highly needed.

Of all the automatic technologies related to SWS, composition is the most challenging one. Several efforts have been made to bring automation into service composition. The method proposed by McIlraith et al [5] is based on AI planning technique. Services are conceived as actions which will change the environment. The PDDL language is adopted to specify each Web Service by its Input, Output, Preconditions, and Effects (IOPE). Situation search method is used to find an action sequence which lead to the goal state.

In Rao's work [6], Web Services, both object service and available services, are represented as Linear Logic (LL) [7] theorems. A LL theorem prover is used to proof the object theorem from available (known) theorems. Implementation of the object service is extracted from the proof in the format of a π-Calculus [8] variation. Ontologies were used to match the sub-class relation of parameters.

2 Background Theory

Web Services are essentially software components. Automatic Service composition is actually the same problem as program synthesis [9]. The theorem proof approach proposed in this paper is based on the deductive program synthesis theory [10]. The overall method of this theory can be outlined in Fig. 1.

Deductive program synthesis theory is based on the observation that proofs are equivalent to programs because each step of a proof can be interpreted as a step of a computation. This transforms program synthesis task into a theorem

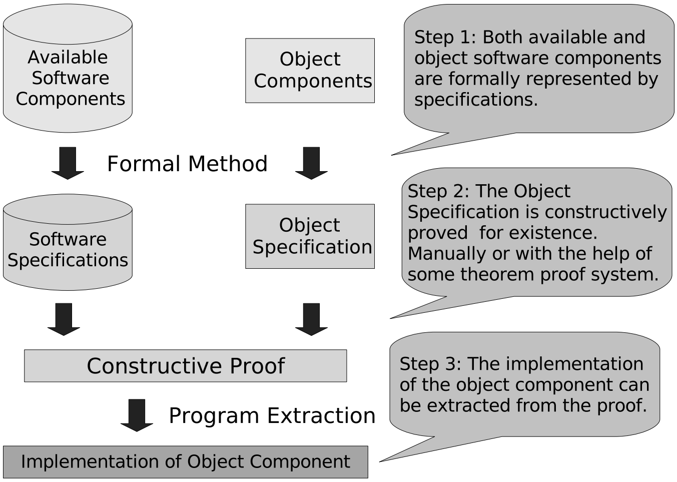

Available Software Components
Object Components
Step 1: Both available and object software components are formally represented by specifications.
Formal Method
Software Specifications
Object Specification
Step 2: The Object Specification is constructively proved for existence. Manually or with the help of some theorem proof system.
Constructive Proof
Step 3: The implementation of the object component can be extracted from the proof.
Program Extraction
Implementation of Object Component

Fig. 1. Typical Workflow of Deductive Program Synthesis method

proof task. An automated theorem prover (ATP), Otter [11], is used in our system to carry out the core proof procedure.

3 Solution Overview

3.1 Functional Specification of SWS

The Inputs, Output, Preconditions, and Result (IOPR) are selected as the 4 elements of SWS's functional specification. Inputs and Output are both subclasses of Parameter which is composed of a name and a type. Preconditions stand for those situations which must exist for the service to be valid. These situations represent the environment.

The Result of a service is composed of two parts: 1) the information transformation performed on the I/O parameters, noted as IO-Relation; and 2) the environment change caused by the service, noted as Effects. The IO-Relation is necessary. For example, a service which increases the input by 1 will be described to have an IO-Relation of "$Successor(input, output)$", and a service which doubles the input will be described to have an IO-Relation of "$Double(input, output)$".

Effects are those things which will happen after the execution of the service. For example, a book selling service might require an input of the bank account the user owns and the book he wants to buy. And the output might be just a confirmation number. The things really happened behind this can be represented as effects, which in this scenario might be the charge of the account and the delivery of the book. So the 'Result' in this example could be encoded as "$Charged(account, book - price)$" & "$Delivered(book, confirm - number)$".

3.2 Translating Service's Functional Specification into FOL Formula

Based on the deductive program synthesis method, the functional specification of all the SWS are translated into FOL formulas before participating in the proof procedure. A formula template is used to do the translation,

$$\forall\ x_1\ x_2\ \ldots\ x_n\ (T_1(x_1)\ \&\ T_2(x_2)\ \&\ \ldots\ \&\ T_n(x_n)\ \&\ P(x_1, x_2, \ldots, x_n)$$

$$\rightarrow\ (\exists\ r\ (T_r(r)\ \&\ Result(x_1, x_2, \ldots, x_n, r)))).$$

where

1. $x_1\ \ldots\ x_n$: represents the input of the service;
2. r: represents the output of the service;
3. $T_*(k)$: means parameter 'k' has the type of 'T_*';
4. $P(x_1, x_2, ..., x_n)$: means "$x_1, \ldots, x_n$" satisfy the precondition 'P' which is defined as an knowledge in the knowledge-base (see Sect. 3.3 for details);
5. $Result(x_1, \ldots, x_n, r)$: Result of the service.

The semantic of the FOL formula can be read as:

> Given the input list of certain types, once the precondition holds, we can get a output which satisfies a certain relationship with the input and be sure of the rise of certain effects.

This FOL formula contains all the necessary functional information of a SWS. The data-type of each parameter (both inputs and output) is specified by a 1-arity predicate with its name as the data-type's name. The "∀ *input* (∃ *output*)" schema means that for every valid input, an output value can be drawn by calling this service (or say applying this formula).

3.3 The Knowledge-Base and the OWL Ontology

The Knowledge-Base (KB) used in our system is composed of FOL rules and predicates. The rules in the KB are knowledge which an automatic agent should know. These rules will participate in the proof procedure together with the formulas which stand for services. The Predicates in the KB defines the terms which could be used to describe the specifications of services.

It is helpful and necessary to use OWL ontology classes to specify the parameter's data-type of the SWS. OWL is based on description logic (DL) [12] which is a subset of FOL. An importing mechanism is defined to guide OWL information into our system. Importing examples are listed in Tab.1. The rules and help services imported are used in the future proof (composing) procedure.

Table 1. Examples of OWL ontology importing

Features	OWL	Importing Result
Class	$Human$	Add Predicate: $Human(1)$, 1 for 1-arity
Inherit	Man inherit $Human$	Add Rule: $\forall\ x\ (Man(x)\ \rightarrow\ Human(x))$
Property	$hasName($ $Domain(Human),$ $Range(Str))$	Add Predicate: $hasName(2)$; Add Help Service "$Human\_Name$": $\forall\ x\ (Human(x) \rightarrow\ (\exists\ r\ (Str(r)\ \&\ hasName(x,r))))$

3.4 Proof and Program Extraction

The proof task is delegated to Otter - An Automated Deduction System. Otter is designed to prove theorems stated in FOL with equality. The object formula (stands for the object service) together with the axioms (stand for the available services) and rules in KB are organized in Otter's input format. If a proof can be found, the implementation of the object service can be extracted from it.

Otter suggested a special tag "$Ans" for the purpose of proof path recording. By adding this tag to the tail of each formula, all the application events of these formula are recorded. Each application event is record by the service name and its parameters. For example, "$Ans(ServiceA(v1,v2,v3))" means that "ServiceA" is called with "v1", "v2" as input and "v3" as output.

The extracting method is straightforward. For example, with the following proof path (ServiceX stands for the object service):

```
\$Ans(Service1(t1, t2)) |
\$Ans(Service2(t2, t3)) |
\$Ans(Service3(t2, t4)) |
\$Ans(Service4(t3, t4, t5)) |
\$Ans(ServiceX(t1, t5))
```

The implementation of the object service (ServiceX) can be extracted as:

```
t1 = input;
t2 = Service1(t1);
t3 = Service2(t2);
t4 = Service3(t2);
t5 = Service4(t3, t4);
output = t5;
```

4 System Implementation

The prototype system, called Service Composer (SC), is under development. Fig. 2 shows the architecture of SC.

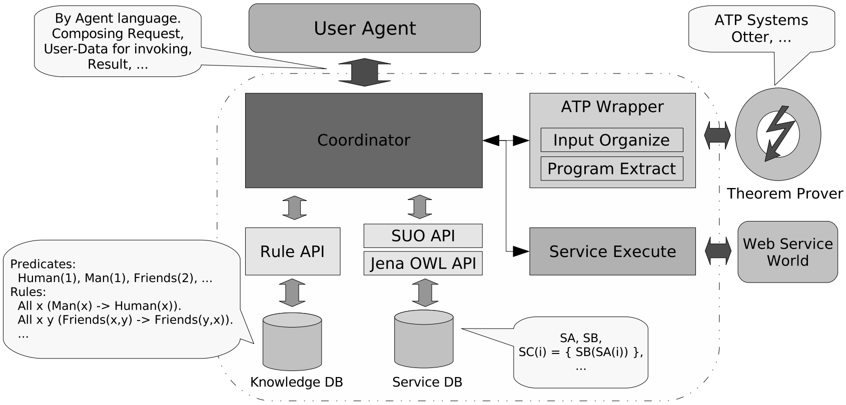
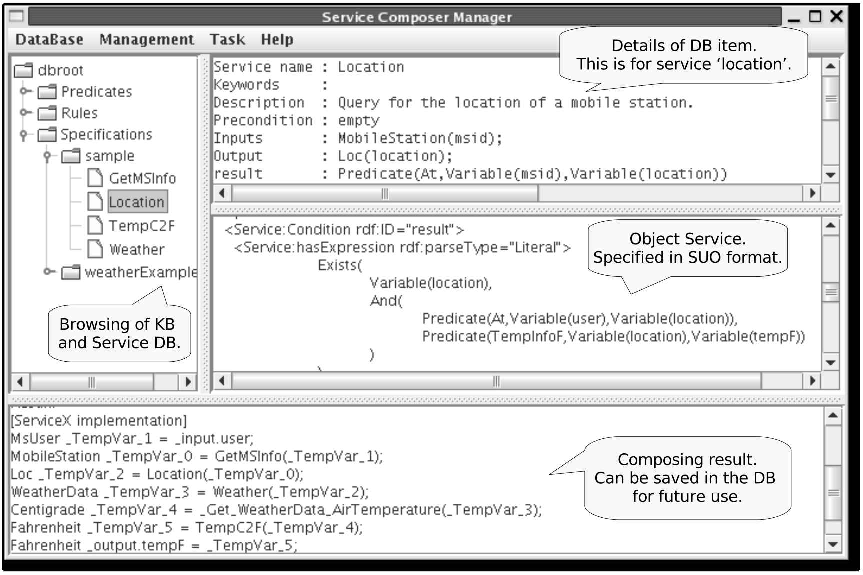


Fig. 2. The Architecture of SC

A demo system (without multi-agent and service invoking feature) has been build to illustrate the feasibility of the system. The core functions including KB, service repository, ontology importing, ATP connecting, and program extracting are implemented. The following screen-shot (Fig. 3) shows the running interface of the system.


Service Composer Manager
DataBase Management Task Help
dbroot
Predicates
Rules
Specifications
sample
GetMSInfo
Location
TempC2F
Weather
weatherExample
Details of DB item. This is for service 'location'.
Object Service. Specified in SUO format.
Browsing of KB and Service DB.
Composing result. Can be saved in the DB for future use.

```
Service name : Location
Keywords     :
Description  : Query for the location of a mobile station.
Precondition : empty
Inputs       : MobileStation(msid);
Output       : Loc(location);
result       : Predicate(At,Variable(msid),Variable(location))
```

```
<Service:Condition rdf:ID="result">
  <Service:hasExpression rdf:parseType="Literal">
          Exists(
                Variable(location),
                And(
                      Predicate(At,Variable(user),Variable(location)),
                      Predicate(TempInfoF,Variable(location),Variable(tempF))
                )
```

```
[ServiceX implementation]
MsUser _TempVar_1 = _input.user;
MobileStation _TempVar_0 = GetMSInfo(_TempVar_1);
Loc _TempVar_2 = Location(_TempVar_0);
WeatherData _TempVar_3 = Weather(_TempVar_2);
Centigrade _TempVar_4 = _Get_WeatherData_AirTemperature(_TempVar_3);
Fahrenheit _TempVar_5 = TempC2F(_TempVar_4);
Fahrenheit _output.tempF = _TempVar_5;
```

Fig. 3. Screen-shot of SC for "air temperature" example

5 Conclusion

The theorem proof approach has its advantages. First, it was based on the FOL, which is already a reliable, mature and well studied mathematics system. It is formal, precise, and concise. The services and knowledges are encoded with their original semantics. This makes them self-explainable.

Second, as a nature of logic system, once the hypothesis stands and the deduce methods are correct, the result is also correct. In this way, the theorem proof approach for service composition used in our system guaranties the correctness of the result. This is the key feature for the automatic mechanism to be used in a composition task because it makes the result believable. Otherwise, a complex algorithm must be designed to prove the correctness of the result when the algorithm itself might became too complex to be proved. The composite service can have firmly and precisely inferred semantics. This semantic can be deduced by the semantics of the services and knowledges which comprise it. It is determinate and explainable.

Third, to the best of our knowledge, this is the first time that IO-Relation is put into consideration while composing web services. Most of the works ([5][6]) put the emphasis on the semantic match of the IO data-types which can only guaranty the type correctness of the result. Many of them do dealing with the effects of service. But as has discussed in Sect. 3.1, the IO-Relation and effects are two different concepts, one for the information transformation feature and one for the environment change feature. None of them are ignorable for the unabridged semantic of the service.

This method depends a lot on the proof ability of ATP systems. Although, due to intensive research (e.g., the German “Schwerpunkt Deduktion” [13]), these systems have gained tremendously in power, they still have weakness. For example, they are not supposed to be good at dealing with recursive datatypes or structures. A proof system with planning support (like the Oyster-Clam system [14]) should be used instead of normal ATPs in this situation.

References

1. Antoniou, G. & Harmelen, F.v.: A Semantic Web Primer. The MIT Press. 2004.
2. Lassila, O. & Swick, R. R.: Resource Description Framework(RDF) model and syntax specification, W3C recommendation. On-line: http://www.w3.org/TR/1999 /REC-rdf-syntax-19990222/, February 1999
3. Dean, M. et al.: OWL Web Ontology Language 1.0 reference, W3C recommendation. On-line: http://www.w3.org/TR/owl-ref/, July 2002.
4. Box, D. et al.: Simple Object Access Protocol (SOAP) 1.1, W3C recommendation. On-line: http://www.w3.org/TR/SOAP/, 2001.
5. McIlraith, S.A.; Son, T.C. & Zeng, H.: Semantic Web Services. IEEE Intelligent Systems, March 2001, 16, 46–53.
6. Rao, J.: Semantic Web Service Composition via Logic-based Program Synthesis. PhD thesis, Norwegian University of Science and Technology, 2004.
7. Girard, J.-Y.: Linear Logic. Theoretical Computer Science, 1987, 50:1-102.
8. Milner, R. & Parrow, J. & Walker, D.: A Calculus of Mobile Processes. Computer Science Department, University of Edinburgh, June 1989.
9. McIlraith, S. & Son, T. C.: Adapting Golog for composition of Semantic Web services. In Proceedings of the 8th International Conference on Knowledge Representation and Reasoning(KR2002), April 2002.
10. Manna, Z. & Waldinger, R.: Fundamentals of deductive program synthesis. IEEE Transactions on Software Engineering, 1992, 18, 674–704.
11. Otter: An Automated Deduction System. On-line: http://www-unix.mcs.anl.gov /AR/otter/.
12. Baader, F. & Calvanese, D. & McGuinness, D. & Nardi, D. & Patel-Schneider, P. & eds.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, 2002.
13. Bibel, W. & Schmitt, P.: Automated Deduction: a Basis for Applications. Kluwer, volume 810.
14. Bundy, A. & Harmelen, F. V. & Horn, C. & Smaill, A.: The Oyster-Clam System. Proceedings of the 10th International Conference on Automated Design, 1990, Springer-Verlag, LNAI 449, 647–648.

A Semantic Rewriting Approach to Automatic Information Providing Web Service Composition

Shenghua Bao[1], Lei Zhang[1], Chenxi Lin[2], and Yong Yu[1]

[1] Department of Computer Science & Engineering
Shanghai Jiao Tong University, Shanghai 200240, P.R. China
{shhbao, zhanglei, yyu}@apex.sjtu.edu.cn
[2] 5F, Beijing Sigma Center No.49, Zhichun Road,
Haidian District Beijing 100080, P.R. China
chenxil@microsoft.com

Abstract. Much work has been done on automatic information providing Web Service discovery and composition, such as the query rewriting approaches proposed by the database community and planning methods in semantic web research. This paper studies the problem of semantic information providing Web Service composition. More specifically, we propose a new method to represent the semantic information providing Web Services in the CARIN language, which seamlessly integrates both database and semantic web technologies. Then, a semantic rewriting based framework and algorithm are proposed to compose the Web Services. Through a case study we show that the new method could find more compositions compared with both query rewriting and planning based Web Service composition methods.

1 Introduction

Based on the open standards (WSDL, SOAP, UDDI) [1], Web Services allow any piece of software to communicate with a standardized XML messaging system. In recent years, a growing number of Web Services have emerged accompanied with the fast developing of Internet. Meanwhile, many more useful solutions have been achieved by composing the existing Web Services into more complex ones. In this paper, we confine ourselves on studying the composition of information providing Web Services. We will use Web Services to denote the information providing Web Services for writing convenience in the rest of the paper.

Manual discovery and composition of Web Services are highly inconvenient and time consuming. Existing work on automatic Web Services composition can be categorized into two classes. One is Web Service composition using query rewriting and the other is semantic Web Service composition using planning.

Query rewriting is initially studied in the database area. It reformulates a user query into new query whose definition refers only to the available views. Methods to solve the automatic Web Services composition problem based on query rewriting techniques have been proposed previously [2,3]. However, the service matching criteria of [2] and [3] are simply based on type match. The semantic information, e.g. equivalent and subclass

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 488–500, 2006.
© Springer-Verlag Berlin Heidelberg 2006

relations between types, are not considered. Thus, the compositions based on semantic information reasoning may be lost in their query rewriting methods.

Planning is used to compose the semantic Web Services. The emerging ontology technology provides the possibility of attaching semantics to each Web Service, by annotating them with respect to service ontologies, e.g. WSMO [4] and OWL-S [5]. The service ontology supplies a core set of markup language constructed for describing the properties and capabilities of Web Services in unambiguous, computer-interpretable form. They facilitate the automation of Web Service tasks. Much work [6,7,8,9,10,11] has been done on automatic semantic Web Service discovery and composition. However, These methods also lose some potential useful Web Service compositions. For example, it is impossible to represent that one services output should be equal to another services due to the restriction of description logic's expression ability [12]. However, in query rewriting, it can be simply done by introducing a free variable and let both these two services outputs be equal to the introduced free variable.

To differentiate from traditional query rewriting, we call the rewriting with respect to a given ontology as semantic rewriting. Semantic rewriting is a special kind of query rewriting. The semantic rewriting related to description logics (DL) [12] is studied with the development of semantic web. Goasdoue and Rousset [13] brought semantic rewriting technology to Semantic Web by studying the problem of answering queries posed through a mediated ontology to multiple information sources whose contents are described as views over the ontology relations. The CARIN language which integrates both database and semantic web technologies seamlessly is proposed as well.

Enlightened by the work of Goasdoue and Rousset, we build the connection between the semantic rewriting and semantic Web Service composition. In this paper, the semantic rewriting based framework and algorithm are proposed to compose semantic Web Services. The key point of the framework is to ignore the differences between input and output of the services and convert the semantic Web Services to conjunctive views of semantic types of a given ontology. Then the semantic rewriting method is applied to find all the rewritings. Finally, the Web Services compositions are obtained from the found rewritings.

The proposed semantic rewriting based method can find more potential compositions compared with existing Web Service composition methods. It can find compositions generated by both the query rewriting methods and planning based methods. It brings the semantic information to the traditional query rewriting methods while breaking the restriction of description logic in planing methods. A case study is conducted to illustrate the effectiveness of the semantic rewriting method in detail.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 provides some preliminaries. Section 4 explores the connection between the semantic rewriting and semantic Web Service composition and proposes the framework of semantic rewriting based Web Service composition. Section 5 describes the corresponding rewriting algorithm. Section 6 explains the algorithm through a case study, and Section 7 discusses the strong and weak points of the proposed framework. Finally, we make some concluding remarks in Section 8.

2 Related Work

Semantic rewriting and semantic Web Service composition are studied and developed previously. Our work explores the connection between these two research areas. Some related work on these two areas is discussed as follows.

2.1 Semantic Rewriting

The semantic rewriting is studied by Goasdoue and Rousset in [13]. They offered CARIN as a family of hybrid logical languages which involved concept description using description logic constructors and rules whose bodies may contain conjuncts that were built using concept descriptions. In addition they exhibited a complete algorithm for a special case of CARIN-$\mathcal{AL}^{+}$. Beeri *et al.* [14] studied rewriting problem of CARIN-$\mathcal{ALN}$. Baader *et al.* [15] studied the problem of rewriting concepts using terminologies, which could be regard as a special case of $\mathcal{FL}_0$.

The semantic rewriting is served as a key component in our composition framework. In this paper, we also proposed a new semantic rewriting algorithm which is designed for the semantic Web Services composition.

2.2 Web Service Composition

Much work has been done on Web Service composition. Query rewriting and planning are two categories of approaches most related to ours.

Lu *et al.* proposed a query rewriting based approach to Web Services synthesis based on the Web Services specification [3]. Thakkar *et al.* extended the inverse rules query reformulation algorithm to generate a universal integration plan to answer user queries [2]. As mentioned in Section 1, the methods in [2] and [3] did not utilize the semantics provided by ontologies.

Semantic Web Services composition using planning has also been well studied and evaluated, e.g. [6,8,9,7,16,17]. Paolucci *et al.* argued that WSDL was not enough to represent the semantic of a Web Service [6] and presented a sample semantic matching algorithm based on DAML-S. Sheshagiri [8] proposed a planner which also made use of services described in DAML-S. Ruoyan Zhang *et al.* [9] proposed the Interface-Matching automatic composition technique that aimed at the automatic generation of complex Web Services by capturing users expected outcomes when a set of inputs are provided. More recently, METEOR-S [7] platform has been established, which provided a comprehensive framework for semantic Web Services and their composition. Benatallah *et al.* [16] proposed an request-rewriting algorithm for Web Service discovery based on hypergraphs. In [17], Sirin *et al.* proposed a HTN planning approach to composite the Web Services.

Different from the above work, in this paper, we take each service as a conjunctive view which contains conjuncts that are built using concept descriptions and then propose the framework and the corresponding rewriting algorithm to perform semantic rewriting based Web Service composition.

3 Preliminaries

To make the paper self-contained, we introduce some preliminaries as follows:

3.1 CRAIN Language

Definition 1. *CARIN is a family of hybrid logical languages in which we can define two kinds of logical sentences over base predicates:*

- *Concept description using description logic.*
  The following is an example of CARIN Concept Descriptions of $\mathcal{AL}$, where A is an atomic concept:

$$C, D \rightarrow A|\top|\bot|\neg A|C \sqcap D|\forall R.C|\exists R.\top$$

- *Rules whose bodies may contain conjuncts that are built using concept descriptions. It has a form of:*$q(\boldsymbol{X}) : - \bigwedge_{k=1}^{n} p_i(\overrightarrow{X_i \sqcup Y_i})$*. where X_i is called existing variable, and Y_i is called free variable, as it does not occur in the left of the equation. An example of CARIN Rules is shown as follows, which are to find the flights that share the same airline:*

$$\begin{aligned} q(X_1, X_2) : - \ & Flight(X_1) \wedge Flight(X_2) \wedge \\ & Airline(X_1, Y_1) \wedge Airline(X_2, Y_1) \end{aligned}$$

The CRAIN language is defined by [18]. Based on the above definition, we can see that CARIN has encompassed many kinds of languages. It will fall back to Terminological Languages if no rule is allowed, Relational Languages with no concept description allowed. If both are allowed, we can derive Hybrid Languages, which is denoted by $CARIN - \mathcal{DL}$ (e.g. $CARIN - \mathcal{AL}$). We will use the CARIN language to denote the language expressive power in the rest of the paper.

3.2 Semantic Rewriting

Let's introduce the definition of semantic rewriting problem proposed by [13].

Definition 2. *Let q be a query defined in $\mathcal{L}_1$ over a given vocabulary, and $\mathcal{V} = v_1, ..., v_k$ be a set of views defined in $\mathcal{L}_2$ over the same vocabulary.*

- *An $\mathcal{L}_3$ **rewriting of** q **using** $\mathcal{V}$ is a query q_v defined in $\mathcal{L}_3$ over the base predicates $v_1, ..., v_k$ such that q_v is subsumed by q modulo $\mathcal{V}$;*
- *An $\mathcal{L}_3$ rewriting q_v using $\mathcal{V}$ is a **maximally contained rewriting** if and only if there is no $\mathcal{L}_3$ rewriting $q_v^{'}$ such that q_v is strictly subsumed by $q_v^{'}$;*
- *An $\mathcal{L}_3$ rewriting q_v using $\mathcal{V}$ is an **equivalent rewriting** if and only if q_v is equivalent to q modulo $\mathcal{V}$.*

Where the $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ are a specific kind of CRAIN language.

3.3 Web Service Composition

First, let's consider a scenario where Web Service composition is needed. Assume John has a trip to Europe on business. He has determined the start time, the start airport and will be accomomodated at a specific inn. What he needs is to find the appropriate airline. John's need can be depicted by OWL-S formally as a Web Service query. We depict it as follows for written convenience:

- $\mathit{def}(RequestService)$: Date, Airport, Inn → Flight

The service query above means that John needs a service whose output is a "Flight", given three inputs "Airport", "Date" and "Hotel".

There are three Web Services available for invoking. The "Flight Service" has input of "Date" and "Airport" and will output the "Flight". The "FlightInfoService" will return the target "Location", given a "Flight". The "HotelInfoService" will return the "Location" and "Grade" of a given "Hotel". Similarly, we depict them as follows:

- $\mathit{def}(FlightService)$: Date, Airport → Flight
- $\mathit{def}(FlightInfoService)$: Flight → Location
- $\mathit{def}(HotelInfoService)$: Hotel → Location, Grade

Let's further assume that "Inn" is a subclass of "Hotel" according to domain ontology $\mathcal{O}$.

$$Inn \sqsubseteq Hotel$$

The problem comes from the scenario above is that none of the three existing Web Services can feed Johns need exactly and we need to compose a new Web Service appropriately based on the existing Web Services.

4 Semantic Rewriting for Semantic Web Service Composition

In this section, we firstly study the connection between semantic rewriting and semantic Web Service composition. Then, we propose the semantic rewriting based framework to perform composition task.

4.1 Semantic Rewriting vs. Semantic Web Service Composition

By comparing semantic rewriting with semantic Web Service composition, we find that they do share many similarities. Table 1 shows the one-to-one mapping between semantic rewriting and semantic Web Service composition.

As we can see from Table 1, both semantic rewriting and semantic Web Service composition are performed with respect to a given ontology $\mathcal{O}$. Both the rewriting query and the Web Service query, existing views and existing Web Services have one to one mapping. Given a query, the semantic rewriting is to find all the contained rewritings against the existing views V and the semantic Web Service composition is to find all the potential compositions to feed the query from existing Web Services S.

Table 1. One-to-one mapping between semantic rewriting and semantic Web Service composition

Semantic Rewriting	**Semantic Web Service Composition**
Ontology $\mathcal{O}$	Ontology $\mathcal{O}$
Rewriting Query q	Web Service Query S_q
Existing Views V	Existing Web Services S

4.2 Semantic Rewriting Framework

Based on the exploited similarities, existing Web Services and the Web Service query can be translated into existing views and rewriting query respectively. Then, the semantic Web Service composition could be translated to semantic rewriting. Semantic Web Service composition and semantic rewriting share the same ontology $\mathcal{O}$ in the whole translation process. The formal description of the semantic rewriting based framework to perform the semantic Web Service composition is presented in Algorithm 1:

Algorithm 1. Semantic Rewriting Framework

```
1 : Translate all the existing Web Services $S$ to existing views $V$;
2 : Translate the Web Services query $S_q$ to rewriting query $q$;
3 : Do the semantic rewriting and generate the rewriting results;
4 : Generate the Web Service compositions from rewriting results and filter
    out the meaningless solutions.
```

Figure 1 graphically depicts the corresponding framework. Each Web Service is depicted by its Input, Output. At first, we omit the difference between the roles of Input and Output in Step 1 and 2. Then, each Web Service is viewed as a typed relation

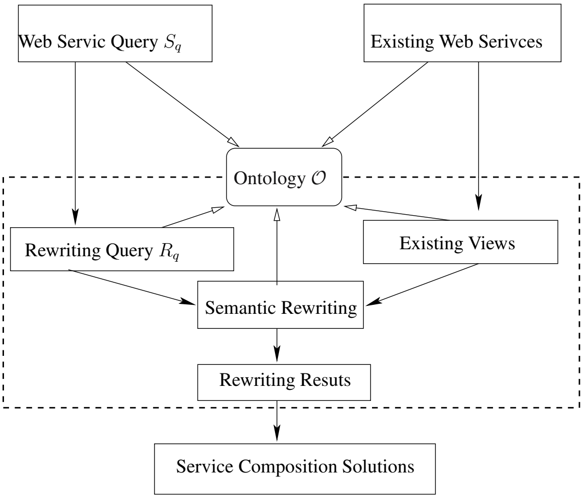


Fig. 1. Framework of semantic rewriting based Web Service composition

$S(x_1, x_2, ...x_n)$, which can be further described as $T_1(x_1) \wedge T_2(x_2) \ldots \wedge T_n(x_n)$. Each conjunct T_i is a concept over ontology $\mathcal{O}$. We perform the semantic rewriting in step 3. Finally, in step 4, the Web Service composition solutions are generated from the rewriting results and we take the difference between roles of Input and Output into consideration by filtering out the meaningless compositions. A composition is meaningless if either of the following condition is satisfied:

- Case 1: **Input/Output Mismatch:** As the Input and Output information is omitted during the semantic rewriting, some Web Service compositions generated from rewriting results may not satisfy the Input/Output restriction required by the Web Service.
- Case 2: **Input/Output Superclass Match:** One services output is another services input and the output type is the super-class of the input type. To guarantee the soundness, the algorithm only allows the cases that the outputs type is equivalent to or subclass of the inputs type.
- Case 3: **Single Connection:** In certain case, omitting a Web Service in a composition, the rest compositions Input and Output can still be satisfied. At the same time, if this Web Service has only one link to the rest of Web Services in the composition, then this composition is redundant and should be filtered out as well.

Till now, the semantic Web Service composition problem has been translated into a semantic rewriting problem. An appropriate rewriting algorithm plays the key role in performing the rewriting based Web Service composition. Next, we discuss the semantic rewriting algorithm in detail.

5 Semantic Rewriting Algorithm for Web Service Composition

In this section, we study the requirement of semantic rewriting task for the semantic Web Service composition and illustrate its differences from the traditional rewriting task. Then we propose the semantic rewriting algorithm for semantic Web Service composition.

5.1 Semantic Rewriting Task

Definition 1 states three kinds of tasks of rewriting, namely, *rewriting*, *maximally contained rewriting* and *equivalent rewriting*. The motivation of traditional query rewriting is usually to find the set of *maximally contained rewriting* of q using V. However, our intention is to find all the potential Web Service compositions. We are mainly concerned with retrieving all the *rewriting*, but not limit to the *maximally contained rewriting* and *equivalent rewriting*, as each rewriting will be translated to a potential Web Service composition. The following is the definition of the semantic rewriting task for Web Service composition.

Task of semantic rewriting:

Given: A DL-based satisfiable ontology O
A satisfiable CARIN-DL query q over O
N satisfiable CARIN-DL views V over O

Return: All the rewriting of q using
$V = \{V_i | i \in [1, N]\}$

In the task, the CARIN-DL query q can be expanded as $q(x_1, x_2, \ldots, x_{a_q}) = \bigwedge_{j=1}^{a_q} T_{qj}(x_j)$ where a_q is the number of variables of query S_q, and T_{qj} is the jth conjunct of query q. The existing view V_i can be expanded similarly, $V_i(x_1, x_2, \ldots, x_{a_i}) = \bigwedge_{j=1}^{a_i} T_{ij}(x_j), i \in [1, N]$, where a_i is the number of variables of view V_i, and T_{ij} is the jth conjunct of view V_i over ontology O.

5.2 Semantic Rewriting Algorithm

In this section, we propose an algorithm of semantic rewriting for Web Service composition. The detail of the algorithm is described in Algorithm 2 as follows:

Algorithm 2. Semantic Rewriting Algorithm

```
3-1: Init: Let process queue $Q$ =null, result set $R = \emptyset$
3-2: For each $T_{qk}, k \in [1, a_q]$, find a subset Setk from existing views: Setk=
     $\{V_i | V_i \in V\ and\ \exists j\ s.t.\ T_{ij} \sqsubseteq T_{qk}\ or\ T_{ij} \sqsupseteq T_{qk}\}$
3-3: Let InitR=$\{r | r = \wedge_{k=1}^{a_q} V_k, V_k \in Setk\}$
3-4: For each rewriting $r$ in InitR, merge the duplicate views, then push it to $Q$
3-5: While($Q \neq$ null) do
     3-5-1: $r$ =shift $Q$, $R = R \cup \{r\}$
     3-5-2: Do internal restriction on $r$ and push new rewriting to $Q$
     3-5-3: Do external restriction on $r$ and push new rewriting to $Q$
```

In Step 3-1, Q is a queue to store the rewriting results to be futher processed and it is initialized to be null. R is where we store the rewriting results and is initialized to be $\emptyset$.

As described in the previous section, $q(x_1, x_2, \ldots, x_{a_q}) = \bigwedge_{j=1}^{a_q} T_{qj}(x_j)$. For a given conjunct T_{qk}, Step 3-2 is to find the existing views that contain a conjunct which is superclass or subclass of T_{qk} over ontology O. In Step 3-3, the initial rewriting results are generated by combining the views selected from each set of $Setk$ and each rewriting r has a form of $\wedge_{k=1}^{a_q} V_k, V_k \in Setk$. Then, in step 3-4, we merge the duplicate views for each rewriting result $r \in InitR$.

Step 3-5 is a loop to find more rewriting results based on the initialized rewriting results that have been found in InitR. In Step 3-5-1, a rewriting r is shifted out from queue Q, and added to the result set. Then, we perform internal restriction and external restriction on r to find new rewritings in Step 3-5-2 and Step 3-5-3 respectively. The new found rewriting is pushed to the queue Q for further process. The internal restriction and external restriction are shown as follows:

- **Internal Restriction:** If two conjuncts T_i and T_j come from two views of a rewriting, $T_i \sqsubseteq T_j$ or $T_i \sqsupseteq T_j$, and at least one of conjuncts is attached to a free variable. Then restrict the free variable to be identical with the existing variable or let the two free variables to be the same.
- **External Restriction:** If one conjunct T_i comes from a view V_i of a rewriting and T_j comes from the existing view V_j that does not appear in current rewriting result yet. $T_i \sqsubseteq T_j$ or $T_i \sqsupseteq T_j$, Then, add the V_j to the existing rewriting and restrict the variable attached to conjuncts T_j to be identical with the variable attached to T_i

In Step 3-2, 3-5-2 and 3-5-3 the reasoning over TBox is needed. Currently, our algorithm could reason based on existing DL reasoning engines such as Racer[19] or FaCT [20].

6 A Case Study of the Framework

To make the algorithm easier to be understood. We illustrate the Web Service composition for the sample proposed in Section 3 step by step according to the algorithm proposed in the last section as follows.

In Step 1: all the existing Web Services S are translated into the existing views V:

- $V_1(x_1, x_2, x_3) : -\ Date(x_1) \wedge Airport(x_2) \wedge Flight(x_3)$
- $V_2(x_1, x_2) : -\ Flight(x_1) \wedge Location(x_2)$
- $V_3(x_1, x_2, x_3) : -\ Hotel(x_1) \wedge Location(x_2) \wedge Grade(x_3)$

In Step 2: the Web Service query $S_q$is also translated into the relational form:

- $q(x_1, x_2, x_3, x_4) : -\ Date(x_1) \wedge Airport(x_2) \wedge Inn(x_3) \wedge Flight(x_4)$

In Step 3-1, we first initialize the queue Q to be null and the result set R to be $\emptyset$. Next, in Step 3-2, the corresponding set $Setk$ is generated for each conjunct of query $q(x_1, x_2, x_3, x_4)$, e.g. ”Date”, ”Airport”, ”Inn” and ”Flight”. Note that when k=3, Set3 is set to be $\{V_3\}$ as ”Inn” is defined to be a subclass of ”Hotel” according to the domain ontology. Then, from Step 3-3 to 3-4, the queue Q is initialized with rewriting(a) $V_1(x_1, x_2, x_4) \wedge V_3(x_3, y_1, y_2)$ and (b) $V_1(x_1, x_2, y_1) \wedge V_3(x_3, y_2, y_3) \wedge V_2(x_4, y_4)$, as shown in Figure.2.

Next, we come to the loop of Step 3-5. Firstly, rewriting(a) $V_1(x_1, x_2, x_4) \wedge V_3$ (x_3, y_1, y_2) is shifted out of the queue Q. Then, two new rewritings (c) and (d) are found through external restriction, as shown in Figure.3.

In the second loop, rewriting (b) $V_1(x_1, x_2, y_1) \wedge V_3(x_3, y_2, y_3) \wedge V_2(x_4, y_4)$ is shifted out of the queue, and one new rewriting (e) is found through internal restriction, as shown in Figure.4(e).

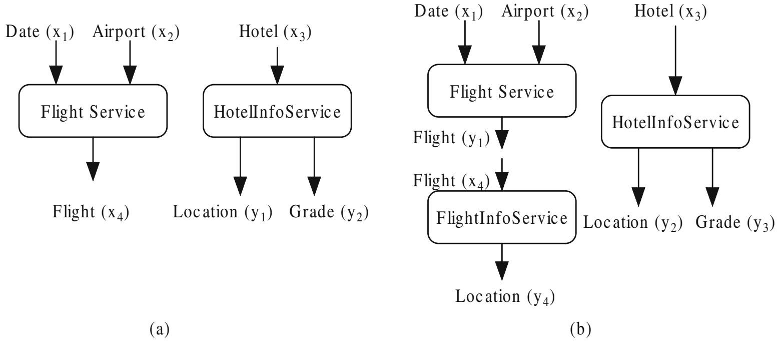


Fig. 2. Rewriting (a) $V_1(x_1, x_2, x_4) \wedge V_3(x_3, y_1, y_2)$ (b) $V_1(x_1, x_2, y_1) \wedge V_3(x_3, y_2, y_3) \wedge V_2(x_4, y_4)$

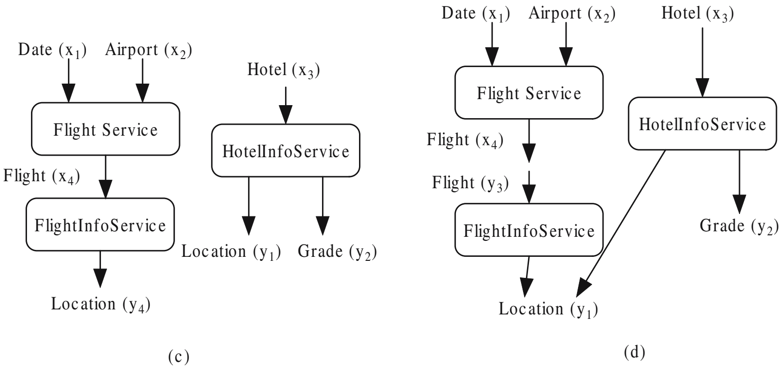


Fig. 3. Rewriting (c) $V_1(x_1, x_2, x_4) \wedge V_3(x_3, y_1, y_2) \wedge V_2(x_4, y_4)$ (d) $V_1(x_1, x_2, x_4) \wedge V_3(x_3, y_1, y_2) \wedge V_2(y_3, y_1)$

In the third loop, rewriting (c) $V_1(x_1, x_2, x_4) \wedge V_3(x_3, y_1, y_2) \wedge V_2(x_4, y_4)$ is shifted out of the queue, and one new rewriting (f) is found through internal restriction, as shown in Figure.4 (f).

The rest of the loops produce no new rewritings and the algorithm terminates with a result set R filled with rewritings (a)-(f).

In step 4, the rewriting (b) and (e) will be filtered out due to Input/Output Mismatch. The rewriting (c) and (d) will be filtered out due to Single Connection. The rewriting (a) and (f) will be presented to the end user as the composition solution.

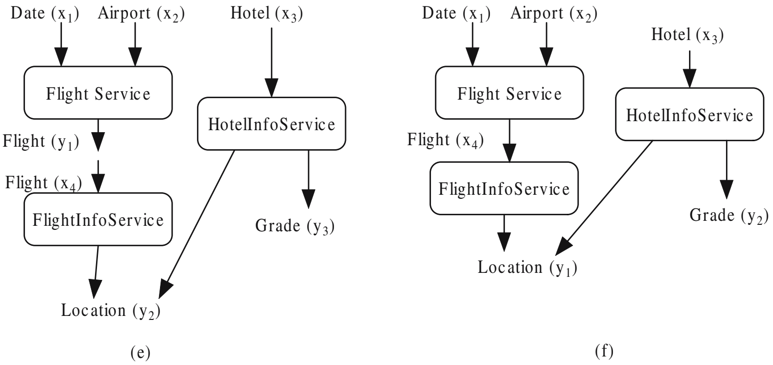


Fig. 4. Rewriting (e) $V_1(x_1, x_2, y_1) \wedge V_3(x_3, y_2, y_3) \wedge V_2(x_4, y_2)$ (f) $V_1(x_1, x_2, x_4) \wedge V_3(x_3, y_1, y_2) \wedge V_2(x_4, y_1)$

7 Discussion

The example illustrated above is really simple for the ease of understanding. Through the example, it is easy to see that our method could produce more potential compositions in comparison with the existing approaches.

- Compared with the previous query rewriting approaches [3,2], our method could deal with more complicated web service types (E.g. $Inn \sqsubseteq Hotel$)with the help of ontology reasoning, to produce more Web Service compositions. Note that we should not use the reasoner to compute the subsumption hierarchy off line. In practice, there will be much more complicated service type expressions. Its impossible to list all in advance.
- Compared with the previous approaches on semantic Web Services compositions, our algorithm can produce more composition results as the Web Services output could further be restricted to be the same. As we can see from the Johns example that the composition derived from rewriting (f) $V_1(x_1, x_2, x_4) \wedge V_3(x_3, y_1, y_2) \wedge V_2(x_4, y_1)$ fits John's need most as this composition will return the "Flight" whose destination is identical with the hotel location. However, the algorithms in the previous work such as [16,6,8], produce the rewriting (a), but do not produce the rewriting (f)where free variable y_1 is involved and can not represented in pure description logic framework.

There are also some restrictions in our method. However, These restrictions do not affect the method employment in an enterprise or organization.

- The Web Service we are mainly concerned with Web Service which is semantic enabled and a unified ontology is required.
- In addition, the algorithm could only deal with the information providing services but not world changing services.
- The time complexity of the rewriting algorithm in Step 3-2 is $O(Avg(|SetK|)^{a_q})$. It is not heavy as the a_q is limited. However, in step 3-5 the time complexity is $O(Avg(|SetK|)^{a_q} * Avg(|Extended\ Web\ Services|))$. It is closely related to the number of extended existing Web Services. Therefore the scalability of our approach would be restricted in global situations.

8 Conclusion and Future Work

In this work, we study semantic rewriting based information providing Web Service composition. The main contributions of the work can be summarized as follows:

- Regarding the semantic Web Services as a kind of conjunctive views in CARIN languages which integrates the datalog and description logics seamlessly.
- Propose a new framework of semantic rewriting based Web Service composition and the corresponding rewriting algorithm.
- Through the case study we also show that semantic rewriting based Web Service composition can find more potential useful compositions in comparison with existing two categories of Web Service composition methods.

The algorithms efficiency will be different with respect to various of ontologies. In future work, we will refine our algorithm for a specific ontology in practice and implement it in our semantic portal [21].

References

1. Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., Weerawarana, S.: Unraveling the web services web: an introduction to soap, wsdl, and uddi. Internet Computing, IEEE **6** (2002) 86–93
2. Thakkar, S., Ambite, J.L., A.Knoblock, C.: A data integration approach to automatically composing and optimizing web services. In: Proceeding of 2004 ICAPS Workshop on Planning and Scheduling for Web and Grid Services. (2004)
3. Lu, J., Yu, Y., Mylopoulos, J.: A lightweight approach to semantic web service synthesis. In: ICDE Workshop, International Workshop on Challenges in Web Information Retrieval and Integration. (2005)
4. Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel, D.: Web service modeling ontology. Applied Ontology **1** (2005) 77 106
5. Martin, D., Burstein, M., Denker, G., Hobbs, J., Kagal, L., Lassila, O., McDermott, D., McIlraith, S., Paolucci, M., Parsia, B., Payne, T., Sabou, M., Sirin, E., Solanki, M., Srinivasan, N., Sycara, K. Technical report, DAML.org (2003) http://www.daml.org/services/owl-s/1.0/.
6. Paolucci, M., Kawmura, T., Payne, T., Sycara, K.: Semantic matching of web services capabilities. In: Proceedings of the First International Semantic Web Conference. (2002) 333–347
7. Patil, A., Oundhakar, S., Sheth, A., Verma, K.: Meteor-s web service annotation framework. In: Proceedings of 13th International World Wide Web Conference. (2004) 553–562
8. Sheshagiri, M., desJardins, M., Finin, T.: A planner for composing services described in daml-s. In: Proceedings of AAMAS Workshop on Web Services and Agent-Based Engineering. (2003)
9. Zhang, R., Arpinar, I., Aleman-Meza, B.: Automatic composition of semantic Web Services. In: Proceedings of The 2003 International Conference on Web Services. (2003)
10. Kumar, A., Srivastava, B., Mittal, S.: Information modeling for end to end composition of semantic web services. In: Proceeding of 2005 International Semantic Web Conference. (2005) 476–490
11. Fensel, D., Bussler, C., Ding, Y., Omelayenko, B.: The web service modeling framework WSMF. In: Proceedings of Electronic Commerce Research and Applications. (2002)
12. Badder, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook. Cambridge University Press (2003)
13. Goasdoue, F., christine Rousset, M.: Answering queries using views: a KRDB perspective for the semantic web. ACM Transactions on Internet Technology (TOIT) **4** (2004) 255 – 288
14. Beeri, C., Levy, A.Y., Rousset, M.C.: Rewriting queries using views in description logics. In: Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems table of contents, ACM Press New York, NY, USA (1997) 99–108
15. Baader, F., Küsters, R., Molitor, R.: Rewriting concepts using terminologies. In Cohn, A.G., Giunchiglia, F., Selman, B., eds.: Proceedings of the Seventh International Conference on Knowledge Representation and Reasoning (KR2000), San Francisco, CA, Morgan Kaufmann Publishers (2000) 297–308
16. Benatallah, B., Hacid, M., Rey, C., Toumani, F.: Request rewriting-based web service discovery. In: Proceedings of 2nd International Semantic Web Conference. (2003)
17. Sirin, E., Parsia, B., Wu, D., Hendler, J., Nau, D.: Htn planning for web service composition using shop2. Journal of Web Semantics **1** (2004)
18. Levy, A., Rousset, M.: Carin: A representation language combining horn rules and description logics. In: Proceeding of European Conference on Artificial Intelligence. (1996) 323–327

19. Haarslev, V., Moller, R.: Racer system description. In: Proceedings of the First International Joint Conference on Automated Reasoning, Springer-Verlag (2001) 701–706
20. Hladik, J.: Reasoning about nominals with fact and racer. In: Proceedings of the 2003 International Workshop on Description Logics (DL2003), Rome, Italy. (2003)
21. Lin, C., Zhang, L., Zhou, J., Yang, Y., Yu, Y.: SPortS: Semantic+Portal+Service. In: ECAI 2004 Workshop on Application of Semantic Web Technologies to Web Communities. Volume 107 of CEUR-WS. (2004)

Web Services Analysis: Making Use of Web Service Composition and Annotation

Peep Küngas[1] and Mihhail Matskin[2]

[1] Norwegian University of Science and Technology
Department of Computer and Information Science
Trondheim, Norway
peep@idi.ntnu.no
[2] Royal Institute of Technology
Department of Microelectronics and Information Technology
Kista, Sweden
misha@imit.kth.se

Abstract. Automated Web service composition and automated Web service annotation could be seen as complimentary methodologies. While automated annotation allows to extract Web service semantics from existing WSDL documents, automated composition uses this semantics for integrating applications. Therefore applicability of both methodologies is essential for increasing the productivity of information system integration. Although several papers have proposed methods for automated annotation, there is a lack of studies providing analysis of the general structure of Web services. We argue that having an overview of general Web services structures would greatly improve design of new annotation methods. At the same time, progress in automated composition has resulted in several methods for automating Web services orchestration. In this paper we propose application of automated composition also for analysing Web services domain. We identify and analyse some general Web services properties and provide their interpretation in an industrial context.

1 Introduction

Automated Web service composition is a field devoted to reduce workforce required for integration of (distributed) information systems. While academic efforts are mainly oriented towards integration at conceptual level, industrial initiatives focus on standards and engagement of existing technologies. In a general context these two approaches are complimentary to each-other.

Before the full power of automated Web service composition could be harnessed, a methodology for mapping existing Web service descriptions in WSDL into conceptual ones should be developed. This methodology would provide a bridge between academic and industrial efforts.

Until now several methods for automated composition [11,19,21,22] have been proposed and demonstrated on simple examples and some scientific applications.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 501–515, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Simultaneously, research on automated annotation [4,5,1,17,13] has resulted in some experiments incorporating machine learning and other techniques. Finally, there is a growing interest in languages and ontologies for describing Web services conceptually. While automated annotation intends to provide methods for extracting semantics from existing Web services, research related to ontologies and Web services has focused on the modelling of Web services. The latter has led to such initiatives as WSMO[1], SWSO[2], OWL-S[3] and WSDL-S[4].

However, if conceptual descriptions of Web services have already been constructed, using them barely for automated composition would be too restrictive—many other things can be done with these descriptions as well. In this paper we discuss some results that can be achieved by analysing conceptual Web service descriptions. We also propose an analysis method which applies automated Web service composition in a novel manner for deducing new facts about a given Web services domain. The method seems to be useful for applying it also in an industrial context.

The work presented in this paper is strongly related to our previous and current work on automated Web service composition [15,9]. After developing a system for automated Web service composition, we were interested to evaluate its performance and applicability in a "real-world" configuration. We were also interested in the current Web services roadmap. More specifically, we would like to know which Web services are available and which are the most common inputs and outputs of current Web services. Our special focus was set to analysis of potential interactions between commercial and governmental Web services.

The rest of the paper is organised as follows. Section 2 explains which Web services we annotated and which language we applied for annotation. Section 3 analyses general differences between commercial and governmental Web services. Section 4 describes how to use automated composition for analysing Web services domains. In Section 5 we present challenges encountered during annotation. Finally, Section 6 reviews related work and Section 7 presents conclusions and elaborates future work.

2 Web Service Annotation

In order to analyse available Web services with our automated composition method, we first annotated manually Web service operations under consideration. For an annotation language we used linear logic (LL) as described by Rao et al [16]. By annotation we mean a process of giving logical names to inputs and outputs of Web service operations. These logical names refer to particular concepts in an ontology and represent the semantics of data, which is exploited by Web services.

[1] www.wsmo.org

[2] www.daml.org/services/swsf/1.0/swso

[3] www.daml.org/services/owl-s

[4] www.w3.org/Submission/WSDL-S

2.1 Formal Annotation Language

Web service operations are described in terms of functionalities and non-functional attributes. The functionalities include inputs, outputs, preconditions, effects and exceptions. The non-functional attributes are classified, according to Rao et al [16], into three categories: consumable quantitative attributes, qualitative constraints and qualitative results. Generally, a required composite Web service can be expressed as the following LL formula

$$(\Gamma_c, \Gamma_v); \Delta_c \vdash ((I \otimes P) \multimap (O \otimes E) \oplus F) \otimes \Delta_r,$$

where both Γ_c and Γ_v are sets of extra-logical axioms representing available *value-added* Web services and *core* Web services. Δ_c is a multiplicative conjunction of non-functional constraints. Δ_r is a multiplicative conjunction of non-functional results.

$I \otimes P \multimap (O \otimes E) \oplus F$ is a functionality description of the required Web service. While I represents a multiplicative conjunction of input parameters of the service, O represents output parameters returned by the service. P and E are multiplicative conjunctions of preconditions and effects, while F is an additive disjunction representing possible exceptions.

Intuitively, the formula can be explained as follows: given a set of available Web services and non-functional attributes, try to find a combination of services that computes O from I as well as changes the world state from P to E. If the execution of the required Web service fails, an exception in F is thrown. Every element in Γ_c and Γ_v is in form

$$\Delta_c \vdash ((I \otimes P) \multimap (O \otimes E) \oplus F) \otimes \Delta_r,$$

where meanings of Δ_c, Δ_r, I, P, O, F and E are the same as described above.

2.2 WSDL Structure

For mapping to annotation language the most essential elements in WSDL are *portType*, *operation*, *message* and *types*. *portType* defines a set of operations, which have input and output messages. While input messages represents operation's input parameters, output messages encapsulate data returned by operations.

Based on the structured information in *portType* LL descriptions can be easily constructed. However, mapping of messages and type definitions is not so simple. This is due to the ambiguity in interpreting this information. For example, if you consider the following WSDL document fragment, you can see that in type definitions of GeoIP and GetGeoIP, there are elements called respectively IP and IPAddress, which refer to the same concept, but have different names.

...

```
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://www.webservicex.net">
      <s:complexType name="GeoIP">
```

```
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="IP" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="CountryCode" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="CountryName" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="GetGeoIP">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="IPAddress" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetGeoIPResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetGeoIPResult" type="tns:GeoIP" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>

  <wsdl:message name="GetGeoIPSoapIn">
    <wsdl:part name="parameters" element="tns:GetGeoIP" />
  </wsdl:message>
  <wsdl:message name="GetGeoIPSoapOut">
    <wsdl:part name="parameters" element="tns:GetGeoIPResponse" />
  </wsdl:message>

  <wsdl:portType name="GeoIPServiceSoap">
    <wsdl:operation name="getGeoIP">
      <wsdl:input message="tns:GetGeoIPSoapIn" />
      <wsdl:output message="tns:GetGeoIPSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
```

...

Thus there are cases where it is not possible to automatically map Web service operations in a WSDL document into LL representation. Therefore mapping from WSDL to LL is currently done mostly manually.

2.3 From WSDL to LL

Generally, Web service operations in WSDL documents can be encoded as follows:

$$\vdash input\_msg \multimap output\_msg$$

For instance, the WSDL document in Section 2.2 can be represented as the following LL specification:

$$\vdash IPAddress \multimap_{getGeoIP} CountryName \otimes ISO3166CountryCode \otimes IPAddress,$$

where *CountryName*, *ISO3166CountryCode* and *IPAddress* refer to particular concepts in an ontology. The LL specification contains a single operation. As one can see we have manually renamed some elements in the WSDL to map them into our ontology—CountryCode to ISO3166CountryCode and IP to IPAddress.

2.4 Web Services Selection

From the list of available commercial Web services we annotated a fraction of the available operations, whose semantics was clear. Altogether we annotated 493 commercial Web service operations. Additionally we developed an ontology for commercial Web services, which consists of 189 relations. The overall commercial Web services domain contains 578 concepts.

For governmental Web services we chose the services from X-Road [12] project, which was initiated by Estonian government in 2001. By March 2005 X-Road had already 41 databases providing services and 354 institutions and companies using the services. The overall number of available Web service operations was at that time 687. We annotated 96 of them. The domain and the developed ontology consists of 595 concepts and 128 relations. The reason of having a larger ontology for commercial Web services is potentially due to the larger heterogeneity of data in this domain. While governmental Web services are centered around queries about citizens and companies, commercial Web services provide a wider set of Web services.

It took in average about one full working day to annotate 100 Web service operations with our primitive tools. Therefore we hope to automate a part of the process to gain higher productivity. Table 1 summarises the number of annotated Web service operations, concepts and relations in developed ontologies. The merged domain indicates that only 24 ((595+578)-1149) concepts were shared between X-Road and commercial ontologies.

Table 1. Annotation overview

Domain	Operations	Concepts	Ontology size
X-Road	96	595	128
Commercial	493	578	189
Merged	589	1149	317

3 General Differences Between Commercial and Governmental Web Services

Our case study identifies the general Web services domain structure as depicted in Fig. 1. The structure was extracted from a previously constructed data flow graph including all annotated Web services. There are 3 components of the domain:

- strict input data
- strict output data
- intermediary data

Strict input data is the data, which only serves as input to any Web service, while strict output data serves solely as output of any Web service. Intermediary

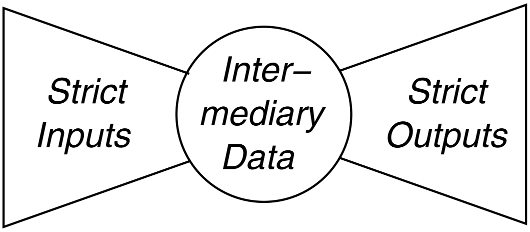


Fig. 1. Web services domain structure

data is presented both in inputs and outputs of Web services. From automated Web services composition point of view strict input and strict output data can exist respectively only in the inputs and outputs of composite Web services. However, intermediary data is the most crucial for the composition—intermediary data allows to compose multiple Web services into a required workflow.

To clarify what we mean by strict inputs, strict outputs and intermediary data, let us consider the Web services domain consisting of the following Web service operations:

$$\vdash IPAddress \multimap_{getGeoIP} CountryName \otimes ISO3166CountryCode,$$

$$\vdash CountryName \multimap_{getCapitalCity} CityName,$$

$$\vdash CityName \multimap_{getPopulationCount} CityPopulationCount,$$

$$\vdash CityName \multimap_{getWeather} Weather,$$

$$\vdash CurrencyCode \multimap_{getRate} CurrencyRate.$$

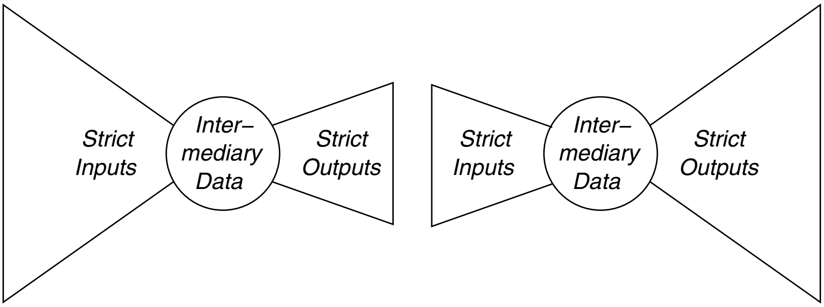
In this domain we have 8 concepts, which are in the following roles:

- strict input data—*IPAddress*, *CurrencyCode*;
- strict output data—*ISO3166CountryCode*, *CityPopulationCount*, *Weather*, *CurrencyRate*;
- intermediary data—*CountryName*, *CityName*.

One may argue here that the concept of strict inputs/outputs is too restrictive since data structures' roles change in time and depend on particular contexts. However, the concept allows to measure the maximum length of automatically composable workflows and to evaluate limitations and applicability of automated Web service composition algorithms as described in Section 4.

Our case study identified a fundamental difference between commercial and governmental Web services domains. While governmental Web services tend to have relatively simple data types for input and more complex data types in outputs, commercial Web services have rather complex data types as inputs and much simpler ones as outputs. This tendency is depicted in Fig. 2. The tendency could be explained by considering the main aims of Web services in these domains. Governmental Web services mostly facilitate access to databases and thus return rich data objects according to simple queries. Commercial Web services, however, are more computation-oriented. They accept rich data structures as input and return compact results of particular computations.

Understanding this difference between commercial and governmental Web services is crucial while developing applications involving Web services from both domains. Furthermore, composite Web services with simple inputs and outputs can be composed by combining Web services from both domains. However, these composite Web services would involve a heavy data transfer between them. Symmetrically composite Web services with rich inputs and outputs can be composed under similar conditions. Anyway, these composite Web services would involve less amount of data transfer between atomic Web services compared to preceding composite Web services.



(a) Commercial Web services (b) Governmental Web services

Fig. 2. Web services' domain structures

Table 2 and Table 3 summarise the number of strict input data, strict output data and intermediary data respectively before and after removing isolated Web service operations from considered domains. When counting the number of relations in developed ontologies, we state explicitly only the number of relations, which represent subclass/superclass relations. Relations, which represent links between Web service operations and data, are not counted.

Table 2. Domain structure overview before removing isolated Web service operations

Data	Commercial	X-Road	Merged
Strict input data	201	20	208
Strict output data	129	205	332
Intermediary data	66	65	123

After this, according to the graph structure [8] representing potential data flow between Web services, we removed respectively 81, 14 and 90 isolated Web service operations from commercial, X-Road and the merged domain. Isolated Web service operations have only strict inputs and strict outputs, respectively for inputs and outputs. For instance *getRate*, from our preceding domain example,

Table 3. Domain structure overview after removing isolated Web service operations

Data	Commercial	X-Road	Merged
Strict input data	156	18	162
Strict output data	97	197	293
Intermediary data	66	65	123

is an isolated operation, since its only input *CurrencyCode* is a strict input and its only output *CurrencyRate* is a strict output.

Since isolated operations are not engaged in data flows, they would not be a part of composite Web services anyway. However, isolated Web services may indicate potential missing Web services, which have to be implemented in order to place them into composite Web services.

4 Automated Composition for Analysis

In this section we describe how we applied an implementation of our method [15] for automated Web service composition to analyse the semantically annotated subset of Web service operations. Our aim is to figure out whether automated Web service composition (in particular, our implementation) is applicable for industrial applications. We would also like to figure out whether the methodology could be applied for analysing existing Web services domains for industrial and academic purposes.

We applied our automated composition method in the following manner. First we included all strict inputs and intermediary data nodes into inputs of the required composite Web service description. For each element from strict outputs and intermediary nodes we applied automated composition such that the output of the required composite Web service consisted of the selected element. An intermediary node in the input part was deleted, if it also existed in the output part. The pseudocode of the algorithm is presented in Fig. 3.

The algorithm takes a set of annotated Web service operations *ops*, strict inputs I, intermediary nodes M and strict outputs O as an input. Then all possible compositions are computed through *compose* and then analysed further by *analyseCompositeServices*. Method *analyseCompositeServices* basically analyses, which composition problems were solved (and which not), which compositions included Web service operations from different domains and which Web service operations mostly occurred in compositions. Additionally composition lengths are analysed. According to that knowledge one can derive most popular Web service operations, possible interaction points between different domains and Web service operations that do not belong to any composition.

We have to emphasise that none of the inputs of the required Web service is mandatory for the required service and they serve as a list of potential inputs for a composite Web service. However, the identified output is mandatory. The composite Web service could have other outputs besides the mandatory ones. Thus the found composite Web services typically involve much less inputs and more outputs than identified initially.

```
Algorithm $AnalyseDomain(ops, I, M, O)$
begin
    $results \leftarrow \emptyset$
    for $\forall output \in M \cup O$
        $inputs \leftarrow I \cup M \setminus output$
        $results \leftarrow results \cup compose(ops, inputs, output)$
    end for
    $analyseCompositeServices(results)$
end $AnalyseDomain$
```

Fig. 3. Automated Web service composition for analysis

To illustrate the algorithm, let us consider the same domain from Section 3. Given that *getRate* was removed from the domain, since it was an isolated operation, we have the following domain topology:

- strict input data—*IPAddress*;
- strict output data—*ISO3166CountryCode*, *CityPopulationCount*, *Weather*;
- intermediary data—*CountryName*, *CityName*.

According to the algorithm we have to apply automated composition to the following Web service descriptions:

$$\vdash IPAddress \otimes CountryName \otimes CityName \multimap_{s_1} ISO3166CountryCode,$$

$$\vdash IPAddress \otimes CountryName \otimes CityName \multimap_{s_2} Weather,$$

$$\vdash IPAddress \otimes CountryName \otimes CityName \multimap_{s_3} CityPopulationCount,$$

$$\vdash IPAddress \otimes CountryName \multimap_{s_4} CityName,$$

$$\vdash IPAddress \otimes CityName \multimap_{s_5} CountryName.$$

Possible compositions for description s_2 are represented by the following operation sequences:

1. getWeather,
2. getCities;getWeather,
3. getGeoIP;getCities;getWeather.

After applying this algorithm to our Web services domains redundant operations were removed from constructed compositions. Redundant operations are operations, which do not contribute to achieving a determined output. They are typically included into composite Web services as side-effects. An example of redundant operations is a Web service operation, which does not have any inputs, but returns current date. Moreover, the current date is not used as an input to other Web service operations. Due to our composition method, redundant operations are often included in resulting composite Web services.

We repeated the procedure, both in forward- and backward-chaining manner, for 3 domains: commercial Web services, X-Road Web services and the merged domain consisting both former domains. While separate analysis of commercial and X-Road Web services allowed to analyse the general characteristics of governmental and commercial Web services, analysis of the merged domain allowed to analyse interactions and potential synergy between commercial and governmental Web services.

For example, it turned out that most popular Web service operations in the merged domain considered either geographical or postal information. Next came operations designed for verifying a postal address and governmental Web service operations. Additionally there were operations for fetching e-mail, processing credit cards and general Internet search. Most popular governmental Web services represented database queries to a business registry.

Throughout our experiments we recorded, which commercial Web service operations were used together with governmental ones in composite Web services. Altogether 25 out of 493 commercial Web service operations were applied together with governmental Web service operations. For 416 tasks in the merged domain 889 solutions were found. The longest composite Web service involved 6 Web service operations. Average composition length was 1.81, mean length was 2.

The maximum length of 6 and mean length of 2 operations in a composite Web service could be due to several factors:

1. small domain size
2. large amount of overlapping Web service operations
3. limitations of the composition algorithm
4. limitations of automated Web service composition in general

Therefore, in order to determine applicability of automated Web service composition in general, we should annotate significantly more Web service operations and repeat the experiments again. Due to the lack of space, only a fraction of results are presented here. The complete set of experimental results and their analysis is presented in [7].

5 Challenges

While annotating Web services and building ontologies we encountered a number of challenges, which either limited our efforts or made in some cases annotation even impossible. A very important factor is the usage of a wide variety of languages in WSDL documents. Although most of the WSDL files were documented in English and the same language was used for naming inputs and outputs, many services contain data in other languages as well. This complicates the extraction of semantics from WSDL files.

Moreover, there is often too few or even misleading information available about Web services and data fields. For instance data field name *country* could be a country name in a particular language or any available country code. There is a general bias not to document data fields in commercial Web services. While X-Road services have mostly data fields commented, only one percent of all available commercial Web services have comments for data fields. Anyway, the situation for service and operation documentation in commercial Web services is much better. In particular, 479 out of 1276 available Web services and 7515 out of 13398 Web service operations in our collection were documented.

Data with the same meaning is encapsulated in different data types. For instance, an address may be represented as a string or as a data type containing fields for each element of an address. Furthermore, sometimes the address contains a country name while in other cases it only represents a street name and house/apartment number.

In summary, the main challenges are as follows:

- different languages
- lack of documentation in WSDL documents
- different data structures with the same meaning
- dynamically changing WSDL documents
- availability of WSDL documents

Therefore, in order to facilitate automated annotation and further usage of annotated Web services, it is desirable to look for alternative descriptions of Web services, like their source code, as done by Sabou [17], or UDDI tModels. Additionally, online dictionaries like WordNet could be exploited to cope with a variety of languages which are used to document WSDL documents. The latter of course requires that there is a way to identify natural languages, which are used within WSDL documents.

6 Related Work

Kim and Rosu [6] are similarly to us concerned with determining generic properties of Web services. However, their main emphasis is set to learn which basic types are mostly used in WSDL descriptions. Based on that statistics they estimate average size of SOAP messages, which are delivered during Web service execution. They also measure average Web service execution time from two different servers.

Several aspects of automated Web service annotation have been considered by research groups. Patil et al [13] present METEOR-S Web service annotation framework. The framework implements new constructs for embedding semantic annotations into existing industry standards. Four kinds of semantics is considered: data, functional, execution and QoS semantics. This contrasts with our approach where we consider just data semantics and embed functional semantics into data semantics. For mapping Web service data types to each-other, initially corresponding XML Schemas are transformed into *SchemaGraphs*. Then linguistic and structural similarity is computed to evaluate the best mapping between existing ontologies and elements in SchemaGraphs. Similarity is measured statistically.

Sabou [17] proposes a semi-automatic method for extracting semantics from software API documentations. The intuition is that, if particular API implements a Web service, then the semantics of API corresponds to the semantics of the Web service. Heß et al [4,5] employ the Naive Bayes and SVM machine learning algorithms to classify WSDL documents according to predefined semantic taxonomies. They allow classification of Web services, their domains and data types. Burstein [1] is concerned with construction of ontology mappings between terms in different Semantic Web services. It is argued that since Web service providers do not use a shared ontology for describing semantically their Web services, automated ontology mapping is required.

Several methods for dynamic composition of Web services have been proposed in recent years. Most of them fall into one of the following two categories: methods based on pre-defined workflow model and methods based on AI planning.

For the methods in the first category, the user should specify the workflow of the required composite service, including both nodes and the control flow and the data flow between the nodes. The nodes are regarded as abstract services that contain search recipes. The concrete services are selected and bound at runtime according to the search recipes. This approach is widely adopted by members of the Information Systems community (in particular, see [2] and [18]).

The second category includes methods related to AI planning and automated theorem proving. They are based on the assumption that each Web service is an action which alters the state of the world as a result of its execution. Since Web services (actions) are software components, the input and the output parameters of Web services act as preconditions and effects in the planning context. After a user has specified inputs and outputs required by the composite service, a process (plan) is generated automatically by AI planners without the knowledge of predefined workflows.

In [11] a modification of Golog [10] programming language is used for automatic construction of Web services. Golog is built on top of situation calculus and has been enriched with some extra-logical constructions like **if**, **while**, etc. Golog also provides an efficient way to handle equivalence relations. Therefore, it is argued that Golog provides a natural formalism for automatically composing services on the Semantic Web.

Sirin et al [20] propose a semiautomatic Web service composition scheme for interactively composing new Semantic Web services. Each time a user selects a new Web service, all Web services, that can be attached to inputs and outputs of the selected service, are presented to the user. In this way a lot of manual search is avoided. Anyway, the process could be fully automated by applying our methodology and if user requirements to the resulting service are known *a priori*.

SWORD [14] is a developer toolkit for building composite Web services. SWORD does not deploy the emerging service-description standards such as WSDL and DAML-S, instead, it uses Entity-Relation (ER) model to specify the inputs and the outputs of Web services. As a result, reasoning is based on the entity and attribute information provided by an ER model.

Gómez-Pérez et al [3] describe another interesting tool for Semantic Web service composition. The resulting service can be exported to an OWL-S specification. In [22] SHOP2 planner is applied for automatic composition of DAML-S services. Other planners for automatic Web service construction include [19,21]. The list is constantly growing.

7 Conclusions and Future Work

In this paper we analysed general differences between commercial and governmental Web services. It turns out that governmental Web services are more data-intensive compared to commercial ones. Having an overview of general characteristics of Web services would greatly improve design of new annotation methods, while knowledge of the presented challenges would contribute to the design of new annotation environments.

We also presented a methodology for analysing Web services' domains through automated Web service composition. The methodology also allows to analyse interactions between Web services' domains and individual Web services. The methodology provides methods for evaluating uniqueness, applicability and other properties of Web services. While our representative set of governmental Web services was selected from X-Road [12] project, commercial Web services were retrieved through Google API. However, due to the lack of space only few analysis results were incorporated to this paper.

As a future work we would like to implement a method for automated annotation of Web services and incorporate it into our annotation tool to facilitate higher productivity and efficiency of the Web services analysis process.

Acknowledgments

This work was partially supported by the Norwegian Research Foundation in the framework of Information and Communication Technology (IKT-2010) program—the ADIS project. The authors would additionally like to thank X-Road project members for their support during the case study and anonymous referees for their comments.

References

1. M. Burstein. Ontology mapping for dynamic service invocation on the Semantic Web. In *AAAI Spring Symposium on Semantic Web Services, Palo Alto, March, 2004*, 2004.
2. F. Casati, S. Ilnicki, L.-J. Jin, V. Krishnamoorthy, and M.-C. Shan. Adaptive and dynamic service composition in eFlow. In *Proceeding of 12th Int. Conference on Advanced Information Systems Engineering (CAiSE 2000), Stockholm, Sweden, June 5–9, 2000*, volume 1789 of *Lecture Notes in Computer Science*, pages 13–31. Springer-Verlag, 2000.

3. A. Gómez-Pérez, R. González-Cabero, and M. Lama. A framework for design and composition of Semantic Web services. In *Proceedings of the First International Semantic Web Services Symposium, AAAI 2004 Spring Symposium Series, March 22–24, 2004*, pages 113–120. AAAI Press, 2004.
4. A. Heß, E. Johnston, and N. Kushmerick. Assam: A tool for semi-automatically annotating semantic web services. In *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004)*, Hiroshima, Japan, 2004.
5. A. Heß and N. Kushmerick. Learning to attach semantic metadata to web services. In D. Fensel, K. Sycara, and J. Mylopoulos, editors, *Proceedings of the 2nd International Semantic Web Conference*, number 2870 in Lecture Notes in Computer Science, pages 258–273, Sanibel Island, Florida, USA, 2003. Springer-Verlag.
6. S. M. Kim and M. C. Rosu. A survey of public Web services. In *Proceedings of 5th International Conference on E-Commerce and Web Technologies, EC-Web 2004, Zaragoza, Spain, August 31–September 3, 2004*, volume 3182 of *Lecture Notes in Computer Science*, pages 96–105. Springer-Verlag, 2004.
7. P. Küngas. *Distributed Agent-Based Web Service Selection, Composition and Analysis through Partial Deduction.* PhD thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway, 2006.
8. P. Küngas and M. Matskin. Web services roadmap: The Semantic Web perspective. In *Proceedings of International Conference on Internet and Web Applications and Services, ICIW'06, Guadeloupe, French Caribbean, February 23–25, 2006.* IEEE Computer Society Press, 2006.
9. P. Küngas, J. Rao, and M. Matskin. Symbolic agent negotiation for Semantic Web service exploitation. In *Proceedings of the Fifth International Conference on Web-Age Information Management, WAIM'2004, Dalian, China, July 15–17, 2004*, volume 3129 of *Lecture Notes in Computer Science*, pages 458–467. Springer-Verlag, 2004.
10. H. J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. B. Scherl. Golog: A logic programming language for dynamic domains. *Journal of Logic Programming*, 31(1–3):59–83, 1997.
11. S. McIlraith and T. C. Son. Adapting Golog for composition of Semantic Web services. In *Proceedings of the Eighth International Conference on Knowledge Representation and Reasoning (KR2002), Toulouse, France, April 22–25, 2002*, pages 482–493. Morgan Kaufmann, 2002.
12. I. Odrats, editor. *Information Technology in Public Administration of Estonia, yearbook 2004.* OÜ Piltkiri, 2005.
13. A. Patil, S. Oundhakar, A. Sheth, and K. Verma. METEOR-S Web service annotation framework. In *Proceedings of the 13th International Conference on World Wide Web (WWW '04), New York, NY, USA, May 17–22, 2004*, pages 553–562. ACM Press, 2004.
14. S. R. Ponnekanti and A. Fox. SWORD: A developer toolkit for Web service composition. In *Proceedings of The Eleventh World Wide Web Conference (Web Engineering Track), Honolulu, Hawaii, USA, May 7–11, 2002*, pages 83–107, 2002.
15. J. Rao, P. Küngas, and M. Matskin. Logic-based Web services composition: From service description to process model. In *Proceedings of the Second International Conference on Web Services (ICWS 2004), San Diego, California, USA, July 6–9, 2004*, pages 446–453, 2004.
16. J. Rao, P. Küngas, and M. Matskin. Composition of semantic web services using linear logic theorem proving. *Information Systems*, 31(4–5):340–360, 2006.

17. M. Sabou. From software APIs to Web service ontologies: a semi-automatic extraction method. In *Proceedings of the Third International Semantic Web Conference (ISWC2004), Hiroshima, Japan, November 7–11, 2004*, 2004.
18. H. Schuster, D. Georgakopoulos, A. Cichocki, and D. Baker. Modeling and composing service-based and reference process-based multi-enterprise processes. In *Proceeding of 12th Int. Conference on Advanced Information Systems Engineering (CAiSE 2000), Stockholm, Sweden, June 5–9, 2000*, volume 1789 of *Lecture Notes in Computer Science*, pages 247–263. Springer-Verlag, 2000.
19. M. Sheshagiri, M. desJardins, and T. Finin. A planner for composing services described in DAML-S. In *Proceedings of the AAMAS Workshop on Web Services and Agent-based Engineering*, 2003.
20. E. Sirin, B. Parsia, and J. Hendler. Composition-driven filtering and selection of Semantic Web services. In *Proceedings of the First International Semantic Web Services Symposium, AAAI 2004 Spring Symposium Series, March 22–24, 2004*, pages 129–136. AAAI Press, 2004.
21. P. Traverso and M. Pistore. Automated composition of semantic web services into executable processes. In *Proceedings of 3rd International Semantic Web Conference, ISWC 2004, Hiroshima, Japan, November 7–11, 2004*, volume 3298 of *Lecture Notes in Computer Science*, pages 380–394. Springer-Verlag, 2004.
22. D. Wu, B. Parsia, E. Sirin, J. Hendler, and D. Nau. Automating DAML-S Web services composition using SHOP2. In *Proceedings of the 2nd International Semantic Web Conference, ISWC 2003, Sanibel Island, Florida, USA, October 20–23, 2003*, 2003.

WWW: WSMO, WSML, and WSMX in a Nutshell*

Dumitru Roman[1], Jos de Bruijn[1], Adrian Mocan[1], Holger Lausen[1,2], John Domingue[3], Christoph Bussler[2], and Dieter Fensel[1,2]

[1] Digital Enterprise Research Institute, Innsbruck, Austria
[2] Digital Enterprise Research Institute, Galway, Ireland
[3] Knowledge Media Institute, the Open University, UK

Abstract. This paper presents, in a nutshell, a unifying framework for conceptually modeling, formally representing, and executing Semantic Web services. We first introduce a conceptual model for representing Semantic Web services and its design principles, then we present a language based on different logical formalisms used to express Semantic Web services that are compliant with our conceptual model. Finally, a high level overview of an execution environment, and its relations to the conceptual model and the language introduced in this paper, are presented.

1 Introduction

Web services [2] - pieces of functionalities which are accessible over the Web - have added a new level of functionality to the current Web by taking a first step towards seamless integration of distributed software components using Web standards. Nevertheless, current Web service technologies (based on specifications like SOAP, WSDL, UDDI, etc.) operate at a syntactic level and, therefore, although they support interoperability (i.e. interoperability between the many diverse application development platforms that exist today) through common standards, they still require human interaction to a large extent: the human programmer has to manually search for appropriate Web services in order to combine them in a useful manner, which limits scalability and greatly curtails the added economic value of envisioned with the advent of Web services [5]. For automation of tasks, such as Web service discovery, composition and execution, semantic description of Web services is required; the usage of ontologies as the basis of such semantic descriptions resulted in a new research area - *Semantic Web Services (SWS)* [9]. In order to provide the basis for SWS, a fully-fledged framework requires three functional layers: a foundational conceptual model, a formal language to provide formal syntax and semantics (based on different

* The work is funded by the European Commission under the projects ASG, DIP, enIRaF, InfraWebs, Knowledge Web, Musing, Salero, SEKT, Seemp, SemanticGOV, Super, SWING and TripCom; by Science Foundation Ireland under the DERI-Lion Grant No.SFI/02/CE1/I13; by the FFG under the projects Grisino, RW^2, SemNet-Man, SeNSE, TSC, OnTourism.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 516–522, 2006.
© Springer-Verlag Berlin Heidelberg 2006

logics in order to provide different levels of logical expressiveness) for the conceptual model, and an execution environment that binds together the several components that use the language to performing various tasks. These functional layers must be provided in order to eventually enable the automation of service.

In this context, this paper gives an overview of such a framework, mainly, it provides a general overview of an ontology for SWS (in Section 2), called Web Service Modeling Ontology (WSMO), a language (in Section 3), called Web Service Modeling Language (WSML), which provides a formal syntax and semantics for WSMO, and an execution environment (in Section 4), called Web Service Modeling Execution Environment (WSMX), which is a reference implementation for WSMO, offering support for interacting with SWS. Section 5 shortly emphasizes related work and concludes this paper.

2 Web Service Modeling Ontology (WSMO)

Web Service Modeling Ontology (WSMO)[11] provides a conceptual model for structuring the semantic annotation of services; it defines ontological specifications for the core elements of SWS. Appropriate frameworks for SWS, need to integrate the basic Web design principles, those defined for the Semantic Web, as well as design principles for distributed, service-orientated computing of the Web. WSMO is therefore based on the following design principles: *Web Compliance* (i.e. uses Web technologies), *Ontology-Based* (i.e. uses ontologies as data model), *Strict Decoupling* (i.e. elements are defined independently from each others), *Centrality of Mediation* (i.e. handling of heterogeneities that naturally arise in open environments), *Ontological Role Separation* (i.e. distinction between the desires of users or clients and available services), *Description versus Implementation* (i.e. differentiation between the descriptions of SWS elements and executable technologies), *Execution Semantics* (i.e. formal execution semantics of reference implementations), and *Service versus Web service* (i.e. differentiation between Web services as a computational entity which is able to achieve a users goal, and services as the actual value provided Web services).

The elements of the WSMO ontology are defined in a meta-meta-model language based on the Meta Object Facility (MOF)[1]. Since WSMO is meant to be a meta-model for Semantic Web services, MOF was identified as the most suitable language/framework for defining the WSMO elements. In terms of the four MOF layers (meta-meta-model, meta-model, model layer, and information layer), the language defining WSMO corresponds to the meta-meta model layer, WSMO itself constitutes the meta-model layer, the actual ontologies, Web services, goals, and mediators specifications constitute the model layer, and the actual data described by the ontologies and exchanged between Web services constitute the information layer (the information layer in this context is actually related to the to the notion of grounding of the semantic descriptions). WSMO provides three main categories to structure semantic descriptions. First, it provides means to describe *Web services*; second, it provides means to describe *user goals* referring

[1] `http://www.omg.org/technology/documents/formal/mof.htm`

to the problem-solving aspect of our framework; and third, it provides means to ensure interoperability between the various semantic descriptions of heterogeneous environments: *ontologies* and *mediators*. For complete item descriptions, every WSMO element is described by a set of non-functional properties.

Goals provide means to characterize user requests in terms of functional and non-functional requirements. For the former, a standard notion of pre- and post-conditions has been chosen and the later provides a predefined ontology of generic properties.

Web service descriptions enrich this by an interface definition that defines access patterns of a service (its choreography) as well as means to express services as being composed from other services (its orchestration). More concretely, a Web service presents: a *capability* which is a functional description of a Web service describing constraints on the input and output of a service through the notions of preconditions, assumptions, post conditions, and effects, and *interfaces* that specify how the service behaves in order to achieve its functionality. A service interface consists of a *choreography* which describes the interface for the client-service interaction required for service consumption, and an *orchestration* which describes how the functionality of a Web service is achieved by aggregating other Web services.

Ontologies provide a first and important means to achieve interoperability between goals and services as well as between various services themselves. By reusing standard terminologies different elements can be either linked directly or indirectly via predefined mapping and alignments. The core elements of an ontologiy include: *concepts* (the basic elements of the agreed terminology for some problem domain), *relations* (model interdependencies between several concepts, respectively instances of these concepts), *instances* (are either defined explicitly or by a link to an instance store), and *axioms* (define complex logical relations between the other elements defined in the ontologies).

Mediators provide additional procedural elements to specify further mappings that cannot directly be captured through the usage of ontologies. Using ontologies provides real-world semantics to our description elements as well as machine processable formal semantics through the formal language used to specify them. The concept of mediation in WSMO addresses the handling of heterogeneities occurring between elements that shall interoperate by resolving mismatches between different used terminologies (data level), communicative behavior between services (protocol level), and on the business process level. A WSMO mediator connects the WSMO elements in a loosely coupled manner, and provides mediation facilities for resolving mismatches that might arise in the process of connecting different elements defined by WSMO. More specifically WSMO defines four types of mediators for connecting WSMO elements: *OO Mediators* connect and mediate heterogeneous ontologies, *GG Mediators* connect Goals, *WG Mediators* link Web services to Goals, and *WW Mediators* connect interoperating Web services resolving mismatches between them.

3 Web Service Modeling Language (WSML)

The Web Service Modeling Language WSML [4] is a language for the specification of different aspects of SWS; it takes into account all aspects identified by WSMO. WSML comprises different formalisms, most notably Description Logics and Logic Programming, in order to investigate their applicability in the context of ontologies and Web services. Three main areas can benefit from the use of formal methods in service descriptions: *ontology description*, *Declarative functional description of goals and Web services*, and *description of dynamics*. So far, WSML defines a syntax and semantics for ontology descriptions. The underlying formalisms which were mentioned earlier are used to give a formal meaning to ontology descriptions in WSML, resulting in different variants of the language, which differ in logical expressiveness and in the underlying language paradigms, and allow users to make the trade-off between provided expressiveness and the implied complexity for ontology modeling on a per-application basis. We briefly describe these variants in the following:

WSML-Core is based on the intersection of the Description Logic $\mathcal{SHIQ}$ and Horn Logic (which is based on Description Logic Programs). It has the least expressive power of all the WSML variants. The main features of the language are concepts, attributes, binary relations and instances, as well as concept and relation hierarchies and datatype support.

WSML-DL captures the Description Logic $\mathcal{SHIQ}(\mathbf{D})$, which is a major part of the (DL species of) OWL.

WSML-Flight is an extension of WSML-Core which provides a powerful rule language. It adds features such as meta-modeling, constraints and nonmonotonic negation. WSML-Flight is based on a logic programming variant of F-Logic and is semantically equivalent to Datalog with inequality and (locally) stratified negation. WSML-Flight is a direct syntactic extension of WSML-Core and it is a semantic extension in the sense that the WSML-Core subset of WSML-Flight agrees with WSML-Core on ground entailments).

WSML-Rule extends WSML-Flight with further features from Logic Programming, namely the use of function symbols, unsafe rules and unstratified negation under the Well-Founded semantics.

WSML-Full unifies WSML-DL and WSML-Rule under a First-Order umbrella with extensions to support the nonmonotonic negation of WSML-Rule.

Several features make WSML unique from other language proposals for the SW and SWS. Amongst them the most important are: one syntactic framework for a set of layered languages (no single language paradigm will be sufficient for all SWS use cases, thus different language variants of different expressiveness are needed); normative, human readable syntax (allows for easier adoption of the language by the users); separation of conceptual and logical modeling (the conceptual syntax allows for easy modeling of ontologies, Web services, goals, and mediators, and the logical expression syntax allows expert users to refine definitions on the conceptual syntax), semantics based on well known formalisms

(WSML captures well-known logical formalisms in a unifying syntactical framework, while maintaining the established computational properties of the original formalisms); and a frame-based syntax (it allows the user to work directly on the level of concepts, attributes, instances and attribute values, instead of at the level of predicates).

These above mentioned features are mainly due to the two pillars of WSML, namely a *language independent conceptual model* for ontologies, Web services, goals and mediators, based on WSMO, and *reuse* of several well-known logical language paradigms in *one* syntactical framework.

4 Web Service Execution Environment (WSMX)

Web Service Execution Environment (WSMX) is an execution environment which enables discovery, selection, mediation, composition and invocation of SWS [6]. WSMX is based on the conceptual model provided by WSMO, being at the same time its reference implementation. WSMX's scope is to provide a testbed for WSMO and to prove its viability as a mean to achieve dynamic interoperability of SWS. In this section aspects of WSMX functionality and WSMX external behavior are briefly presented.

WSMX functionalities can be classified in two main categories: first is the functionality required to support the operations (e.g. discovery or invocation) on SWS and second, the additional functionality coming from the enterprise system features of the framework. In the first case, the overall WSMX functionality can be seen as an aggregation of the components' functionalities, which are part of the WSMX architecture. In the second case, WSMX offers features such as a plugging in mechanism that allows the integration of various distributed components, an internal workflow engine capable of executing formal descriptions of the components behavior or a resource manager that enables the persistency of WSMO and non-WSMO data produced during run-time.

WSMX external behavior is described in terms of so-called entry points which represents standard interfaces that enable communication with external entities. There are four mandatory entry points that have to be available in each working instance of the system. Each of these entry points triggers a particular execution semantics which on its turn, selects the set of components to be used for that particular scenario:

One-way goal execution. This entry point allows the realization of a goal without any back and forth interactions. In this simplistic scenario the requester has to provide a formal description of its goal in WSML and the data required for the invocation and the system will select and execute the service on behalf of the requester. The requester might receive a final confirmation, but this step is optional.

Web Service discovery. A more complex (and realistic) scenario is to only consult WSMX about the set of Web services that satisfy a given goal (the selection might take place later, e.g. at the requester side). This entry point implies an synchronous call, the requester provides a goal and WSMX return a set of matching Web services.

Send message. After the decision on which service to use was already made, a conversation involving back and forth messages between the requester and WSMX can start. The input parameter is a WSML message that contains a set of ontology instances and references to the Web service to be invoked and to the targeted choreography (if it is available).

Store entity in the registry. This entry point provide an interface for storing WSMO entities (described in WSML) in the repository.

It is important to note that all the incoming and outgoing messages are represented in WSML and they are either fragments of WSMO ontologies or WSMO entities (Web services, goals, mediators, or ontologies). That is, only WSML is used as WSMX internal data representation, and all the necessary adaptations operations to and from other representation formats are handled by an adapters framework.

WSMX architecture consists of a set of loosely decoupled components[2] and follows the fundamental principles of a Service Oriented Architecture (SOA). Even if WSMX provides default implementations for all the components in the architecture, self-contained components with well defined functionalities can be easily plugged-in and plugged-out at any time.

5 Conclusions and Outlook

SWS constitute one of the most promising research directions to improve the integration of applications within and across enterprise boundaries. Besides WSMO, several other approaches to SWS have been proposed[3]. Amongst them, the most important are OWL-S [12], SWSF[1], IRS-III[7], and WSDL-S[10]. However, compared to the WSMO approach highlighted in this paper, none of these approaches tackle, in a unifying manner, all the aspects of a framework for SWS. In this context, WSMO provides the conceptual and technical means to realize SWS, improving the cost-effectiveness, scalability and robustness of current solutions, WSML provides a formal syntax and semantics for WSMO by offering different variants based on different logics in order to provide different levels of logical expressiveness, and WSMX provides a reference implementation for WSMO and interoperation of SWS.

In total, the framework highlighted in this paper sets a solid basis for solving the research issue on SWS. We refer the reader to the WSMO web site[4], where several research results based on WSMO (including primers, tutorials, use cases, tools, theoretical results and investigations on how to apply the framework presented in this paper, etc.) can be accessed. Moreover, with the shift towards

[2] For more details we refer the reader to the WSMX code base at Sourceforge (http://sourceforge.net/projects/wsmx).

[3] We refer the reader to [3] for a detailed discussion on the SWS approaches.

[4] `http://www.wsmo.org/`

service orientation, WSMO, WSML, and WSMX form the basis for the Semantically Enabled Service-Oriented Architectures [8]. With the WSMO Submission[5] to W3C, and the formation of the OASIS Semantic Execution Environment Technical Committee[6], the framework presented in this paper is expected to have a high impact on the standardization activities around SWS.

References

1. Semantic Web Services Framework. SWSF Version 1.0. Available from `http://www.daml.org/services/swsf/1.0/`, 2005.
2. G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services: Concepts, Architecture and Applications*. Springer Verlag, 2004.
3. D. Roman, J. de Bruijn, A. Mocan, I. Toma, H. Lausen, J. Kopecky, D. Fensel, J. Domingue, S. Galizia, and L. Cabral. Semantic Web Services - Approaches and Perspectives. In P. Warren J. Davies and R. Studer, editors, *Semantic Web Technologies: Trends and Research in Ontology-based Systems*. John Wiley & Sons, 2006.
4. J. de Bruijn, editor. *The Web Service Modeling Language WSML*. 2005. WSMO Final Draft D16.v0.21. Available at `http://www.wsmo.org/TR/d16/d16.1/v0.21/`.
5. D. Fensel and C. Bussler. The Web Service Modeling Framework WSMF. *Electronic Commerce Research and Applications*, 1(2), 2002.
6. A. Haller, E. Cimpian, A. Mocan, E. Oren, and C. Bussler. WSMX - A Semantic Service-Oriented Architecture. *International Conference on Web Services (ICWS 2005)*, July 2005.
7. J. Domingue, L. Cabral, F. Hakimpour, D. Sell, and E. Motta. IRS-III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services. In *Proceedings of the Workshop on WSMO Implementations (WIW 2004)*. CEUR, 2004.
8. M. L. Brodie, C. Bussler, J. de Bruijn, T. Fahringer, D. Fensel, M. Hepp, H. Lausen, D. Roman, T. Strang, H. Werthner, and M. Zaremba. Semantically Enabled Service-Oriented Architectures: A Manifesto and a Paradigm Shift in Computer Science. Technical Report TR-2005-12-26, 2005. Available from `http://www.deri.at/fileadmin/documents/DERI-TR-2005-12-26.pdf`.
9. S. McIlraith, T. Son, and H. Zeng. Semantic Web services. In *IEEE Intelligent Systems (Special Issue on the Semantic Web)*, 2001.
10. J. Miller M. Nagarajan M. Schmidt A. Sheth R. Akkiraju, J. Farrell and K. Verma. Web Service Semantics - WSDL-S. Technical report, 2005. Available from `http://lsdis.cs.uga.edu/library/download/WSDL-S-V1.html`.
11. D. Roman, U. Keller, H. Lausen, R. Lara J. de Bruijn, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel. Web Service Modeling Ontology. *Applied Ontology*, 1(1):77–106, 2005.
12. The OWL Services Coalition. OWL-S 1.1. Available at `http://www.daml.org/services/owl-s/1.1/`, 2004.

[5] `http://www.w3.org/Submission/2005/06/`

[6] `http://www.oasis-open.org/committees/semantic-ex/`

Automatic Generation of Service Ontology from UML Diagrams for Semantic Web Services

Jin Hyuk Yang and In Jeong Chung*

Dept. of Computer & Information Science, Korea Univ., Korea
{grjinh, chung}@korea.ac.kr

Abstract. We present in this paper the methodology for automatic generation of OWL-S service model ontology along with results and issues. First we extract information related to atomic services and their properties such as IOPE from UML class diagram, and retrieve information related to composition of services from UML state-chart diagram. Then XSLT applications utilize the acquired information to generate the OWL-S service model ontology through the predefined mappings between OWL-S constructs for composite services and UML state-chart primitives. For the justification of generated service ontology several validation checks are performed. Our service ontology generation method is fully automatic and effective in that it is performed in familiar environment to developers and information needed to generate service ontology is provided necessarily during service development. It is also noticeable to facilitate representing the condition with GUI rather than complex language like OCL.

Keywords: Ontology, Semantic Web, OWL-S, state-chart, and UML.

1 Introduction

Semantic web services, often called as intelligent web services, first introduced in [2], enables web services to be intelligent using ontologies which play an important role as metadata for inferencing in semantic web. Such intelligent web services include automatic services' discovery, execution, composition, and interoperation, which are the goals of OWL-S[5]. OWL-S is service ontology written in OWL[6], adopted by W3C as standard. In this paper we present the methodology for automatic generation of OWL-S service ontology along with results. In particular we focus on the OWL-S service model ontology among three OWL-S ontologies(service profile, service model and service grounding), since crucial information on how to interoperate with other services is described within the service model ontology. This kind of information is essentially required to enable the intelligent web services like automatic web services composition.

For automatic generation of OWL-S service model ontology we extract information related to atomic services and their properties such as IOPE(Input,

* Corresponding author.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 523–529, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Output, Precondition, and Effect)s from the UML class diagram, and retrieve information related to composition of services from UML state-chart diagram. Then XSLT applications utilize the acquired information to generate OWL-S service model ontology through the defined mappings in section 3 between OWL-S constructs for composite services and UML state-chart primitives. For the justification of generated ontology we performed a few of validation checks available.

Our service ontology generation method is not only fully automatic but also effective in that it is performed in familiar environment to developers and information needed to generate service ontology is necessarily provided during service development. This familiar environment means they use only UML and don't need to use OWL-S primitives to generate OWL-S ontology. In addition we propose the method for modeling OWL-S condition expression with GUI in UML diagram instead of complex language like OCL. Detail explanation on this issue is addressed again in subsection 3.2.

2 Related Works

Main idea of using ontology is to pursue automation and intelligence via reasoning on metadata about resources. However the task of creating ontology is time-consuming and difficult as indicated in [9]. Therefore automatic and effective ontology creation is very important.

It can be found in [13] as related work on creating OWL-S service ontology. [13] uses service's auto-generated WSDL[1] document and annotates it. [14] uses UML activity diagram to generate service's BPEL4WS[3] specification. However [13] is semi-automatic, therefore bothers the developers. [14] is similar with our approach; UML activity diagram and BPEL4WS rather than UML state-chart diagram and OWL-S. Comparisons between OWL-S and BPEL4WS are described in [7] and [8]. And the cause of using state-chart instead of activity diagram is state-chart diagram has not only well-defined semantics but also basic flow constructs such as sequence, conditional branching, structured loops, concurrent threads and synchronization primitives as in most process modeling languages[25]. According to [25], these features facilitate applying formal manipulation techniques to state-chart model and guarantee that state-chart can be adapted to other web service modeling languages such as BPEL4WS and WSCI[4].

Most remarkable approach is [27], which automatically generates OWL-S ontology from WSDL document through the mappings between OWL-S and WSDL. However [27] enforces manual processing for completing the OWL-S service model ontology when it contains more than one atomic process(i.e., when the service is composite process). In fact this is due to the expressiveness of WSDL. Therefore our research can be regarded as complementary approach to [27]. Moreover we present the way of expressing the condition with GUI instead of complex language like OCL.

3 Automatic Generation of OWL-S Service Ontology

3.1 Motivation and Design Principle

Service ontologies have to be created in order to realize the semantic web services. If it is expensive and time-consuming to create the service ontologies, it is an obstacle for populating ontologies. Therefore it is desirable to create service ontologies in automatic and effective. Even some tools such as [15] and [16] provide ontology editing environment, they enforce modelers(developers) to understand OWL-S. Moreover extra time and efforts are needed in modeling ontologies.

– Principle: Service ontology must be generated in an automatic and effective manner.

To populate the service ontologies necessarily needed in realization of semantic web services, it is highly desirable to generate the service ontologies in automatic as services' WSDL documents are automatically generated in Java and .NET environment. In addition it must be performed in easy and familiar environment to developers, since most developers(creator of service ontologies) are unfamiliar with OWL-S.

To reflect the principle above we consider XSLT and UML to generate the service ontology. UML is widely adopted in software engineering as GUI standard for modeling, which is familiar with most developers. Once necessary information are extracted from UML diagrams, XSLT applications automatically transform the UML diagrams into OWL-S specification using the rules defined in the next section.

3.2 Mapping Definitions Between OWL-S Constructs for Composite Services and UML State-Chart Primitives

In this section we address several rules embodied in XSLT application which is used to generate OWL-S service model ontology. These rules are based on the mappings between OWL-S constructs for composite services and UML state-chart primitives.

We consider two separate processes in generating OWL-S service model ontology: first process of generating atomic services and their IOPE-related attributes, second process of generating information related to composite services. In order words information related to services' attributes is extracted from UML class diagram in the first process, and composition information related to composite service is extracted form UML state-chart diagram in the second process. The reason why we extract different information from different UML diagrams is that UML class diagram is good to describe the atomic services, their attributes and relationships with other atomic services while UML state-chart diagram is good to model services' behavior and allows describing composite service composed of other composite services. Namely UML class diagram is not suitable to do what UML state-chart does and visa versa.

In the following, we only define the mappings between OWL-S constructs for composite service and UML state-chart diagram primitives, since mappings between primitives of UML class diagram and OWL-S's attribute-related information are simple as well as already defined in [10,11,12].

- Sequence: OWL-S Sequence is a construct for specifying the sequence of services. It is defined as stereotyped transition.
- Split and Split+Join: OWL-S Split and Split+Join are constructs for modeling synchronization. They are defined using Fork/Join primitive.
- Choice and AnyOrder: OWL-S AnyOrder and Choice are constructs for modeling for selections. They are defined using Choice primitive.
- If-Then-Else: OWL-S If-Then-Else is a construct for modeling conditional branching. It is defined using Choice for branching. In addition stereotyped class and dependency are used.
- Iterate, Repeat-While and Repeat-Until: OWL-S Iterate and its subclasses Repeat-While and Repeat-Until are structured loop constructs. Repeat-While and Repeat-Until are defined as combinations of mapping definition used for OWL-S If-Then-Else(for specifying condition) and stereotyped transition primitive. Repeat-Until is defined at same way.

It is noticeable in the mapping definitions how OWL-S If-Then-Else construct is mapped to UML state-chart diagram primitives. The reason of using class and dependency beyond UML state-chart diagram primitives is that we decide to use GUI for representing condition. Allowed languages for expressing condition in OWL-S include SWRL[17], RDF[18], KIF[19] and PDDL[20]. Among these specifications, we choose SWRL, since it is not only layered on top of OWL but also considered as candidate standard for rule expression by DAML.org. Atoms of SWRL can be created using unary predicates(classes), binary predicates(properties), equalities and inequalities. These SWRL atoms are children of ruleml:_body and ruleml:_body which have ruleml:_imp as their parent component in RuleML[21]. Among lots of constructs for various SWRL atoms, we choose three first of all: classAtom, individualPropertyAtom and buitinAtom. However others can be similarly modeled in our approach.

Again here is reason of using class and dependency for modeling OWL-S condition expressed with SWRL. Condition(predicate) name and information related to arguments and their types are needed to specify the condition using three above SWRL atoms. Some of our considerations on describing necessary information for specifying condition in UML state-chart diagram include use of OCL and direct representation of SWRL expression within note section. However using OCL or specifying note is enforcing developers to understand OCL and SWRL and this approach is not considered as automatic, because represented condition expression in OCL or SWRL must be parsed and manipulated again. In order to overcome this problem, we decide to use class and dependency. Stereotyped dependency is used for representing SWRL atom type while stereotyped class is used for describing SWRL atom name. Attributes are used to describe the condition's arguments and their types within the class. This approach makes it easy to express condition with GUI in UML state-chart diagram and allows transformation to be automatic and simple.

4 Case Study Implementation

4.1 Simple Scenario and Design

As simple scenario we choose and adapt the one introduced in [2], which introduce the semantic web service first. This is about travel service. Someone wants to travel one place to other one by using airplane or automobile. If driving time from source place to destination takes greater than 3 hours, then she/he wants to take a flight. Otherwise she/he is going to rental a car. Furthermore, we assume entire travel service is composed two separate steps: Once deciding transportation means, then selecting accommodation. There are two possible transportation means: airplane and automobile.

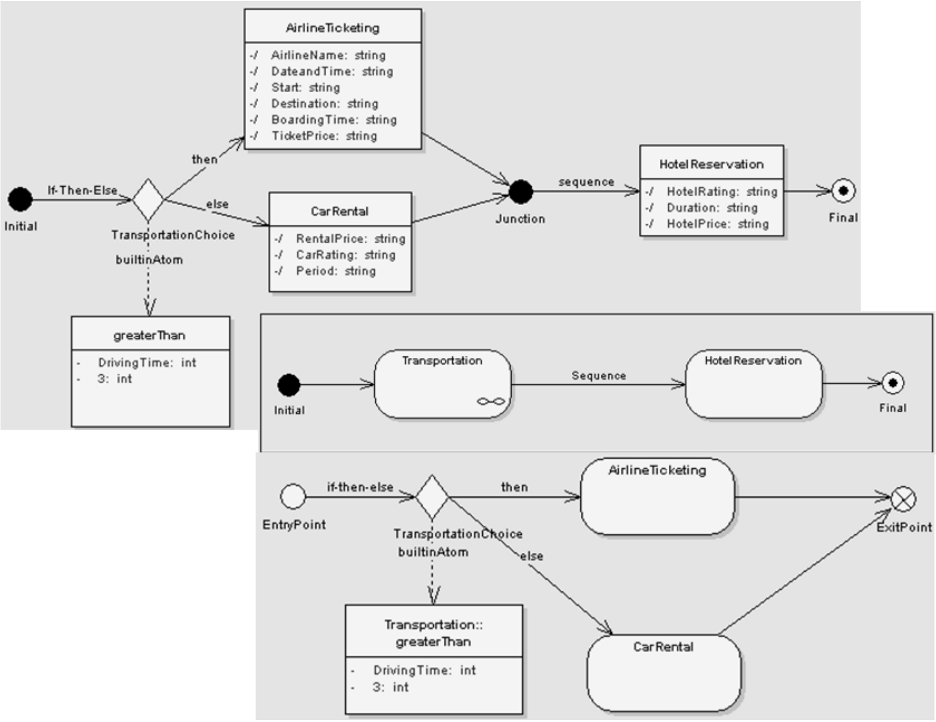


Fig. 1. Class and state-chart diagram for travel service

Top part of Fig. 1 depicts our scenario. We use class diagram for logically modeling travel service which is composed of three atomic services: AirlineTicketing, CarRental and HotelReservation. Middle part of Fig. 1 tells entire service is composed of one composite service(Transportation) and atomic service(HotelReservation). It also depicts entire service must be executed in sequence. Bottom part of Fig. 1 is expansion of the composite service: Transportation. Note how the condition assumed in our scenario is modeled with UML primitives. The condition GUI tells if driving time is greater than 3 hours(can be expressed as greaterThan(DrivingTime, 3)), then AirlineTicketing service will be used, otherwise CarRental service.

4.2 Transformation and Validation

UML class diagram and state-chart diagram are exported to two separate XMI files respectively, then two XSLT applications(written with help of [26]) produce output files. Finally, the OWL-S service model ontology is produced(due to the space limit, we omit the transformation algorithm, generated OWL-S service model ontology, and validation results).

We validate and confirm generated service ontology in several steps, since OWL-S validator provided by standard organization like W3C is not available. First we use the site[22] available in W3C for RDF-level test. As W3C does not support beyond RDF, we should use other sites for validating our generated ontology. Fortunately a couple of OWL validators are available. Among them we use [23] and verified. In respect to OWL-S validation we can find [24]. However they don't support OWL-S version 1.1. Moreover their OWL-S validator does not support all the OWL-S constructs for composite service; they support only Sequence, Unordered and Split. Therefore we validated and confirmed our generated ontology except If-Then-Else construct part.

5 Conclusion and Future Work

We propose the method for generating OWL-S service model ontology where service's behavior is described. Our approach of generating service ontology is fully automatic as well as effective in that it is performed in familiar environment and information needed to generate service ontology is provided necessarily during service development. Another contribution may be representing condition using GUI rather than complex language like OCL.

As future work we consider the way of expressing with GUI complex condition where more than one predicate are represented. Simply several dependency and class types can be used to express the complex condition. However we may encounter a case where we should carefully consider evaluation order of individual condition of the complex condition expression.

References

1. WSDL, http://www.w3.org/TR/2004/WD-wsdl20-primer-20041221/
2. Sheila A. Mcllraith, Tran Cao Son, Honglei Zeng, Semantic Web Services, IEEE Intelligent Systems, pp.46-53, 2001.
3. BPEL, http://www-128.ibm.com/developerworks/library/specification/ws-bpel/

4. WSCI, http://www.w3.org/TR/wsci/
5. OWL-S, http://www.daml.org/services/owl-s/1.1/
6. OWL, http://www.w3.org/TR/owl-features/
7. http://www.daml.org/services/owl-s/1.1/related.html
8. http://www.daml.org/services/daml-s/0.9/survey.pdf
9. Michele Missikoff, Roberto Navigli, Paola Velardi, The Usable Ontology: An Environment for Building and Assessing a Domain Ontology, ISWC 2002, LNCS 2342, pp.39-53, 2002.
10. Cranefield S., Purvis M., UML as a Ontology Modeling Language, Proc. Of the Workshop on Intelligent Information Integration, 16th Int. Joint Conference on AI(IJCAI-99), 1999.
11. Cranefield, S., Haustein, S., and Purvis, M., UML-Based Ontology Modelling for Software Agents, Proceedings of the Workshop on Ontologies in Agent Systems, 5th Internal Conference on Autonomous Agents, pp.21-28, 2001.
12. K. Baclawski, M. Kokar, P. Kogut, L. Hart, J. Smith, W. Holmes, J. Letkowski, M. Aronson, P. Emery, Extending the UML for Ontology Development, SOSYM 2002, Software System Model(2002) Vol.1, pp.1-15, 2002.
13. Andreas H., Eddie J., and Nicholas K., ASSAM: A Tool for Semi-automatically Annotating Semantic Web Services, ISWC 2004, LNCS 3298, pp. 320-334, 2004.
14. Keith Mantell, From UML to BPEL: Model Driven Architecture in a Web Services world, http://www-128.ibm.com/developerworks/webservices/library/ws-uml2bpel/
15. Protege, http://protege.stanford.edu
16. http://staff.um.edu.mt/cabe2/supervising/undergraduate/owlseditFYP/OwlSEdit.html
17. SWRL, http://www.daml.org/2004/04/swrl/
18. RDF, http://www.w3.org/TR/2004/REC-rdf-concepts- 20040210/
19. Knowledge Interchange Format: Draft proposed American National Standard(dpans). Technical Report 2/98-004, ANS, 1998.
20. M.Ghallab et al., Technical Report, report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, 1998.
21. RuleML, http://www.ruleml.org/
22. RDF validator, http://www.w3.org/RDF/Validator/
23. ConsVISor, http://www.vistology.com/consvisor/
24. http://www.mindswap.org/2004/owls/validator
25. Lianzhao Zeng, Boualem Benatallah, Marlon Dumas, Jayant Kalagnanam, Quan Z. Sheng, Quality Driven Web Services Composition, WWW2003, pp.411-421, 2003.
26. Stylus Studio XML, http://www.stylusstudio.com/
27. Massimo Paolucci, Naveen Srinivasan, Katia Sycara, Takuya Nishimura, Towards a Semantic Choreography of Web Services: form WSDL to DAML-S, In Proceedings of First Internal Conference on Web Services(ICWS'03), pp.22-26, 2003.

A Composition Oriented and Graph-Based Service Search Method

Xiaoqin Xie[1], Kaiyun Chen[2], and Juanzi Li[3]

[1] College of Computer Science and Technology,
Harbin Engineering University, 150001 Harbin, China
xiexiaoqin@tsinghua.org.cn
[2] College of Mechanics and Electronics Engineering,
Harbin Engineering University, 150001 Harbin, China
chenkaiyun@tsinghua.org.cn
[3] Computer Science and Technology Department, Tsinghua University,
100084 Beijing, China
ljz@keg.cs.tsinghua.edu.cn

Abstract. When there do not exist the directly matched services but exist several services in the repository that can be combined to meet the requirements, how to discovery the multiple services and their composition relations are the critical issues. This paper proposes a composition-oriented and AND/OR graph-based service search method named as CoSA, which can automatically search the composition relation graph in concepts level and composition plan in implement level. Composition operator and domain characteristics are reflected in the heuristic functions. CoSA decreases the service search space and improves the search effectiveness. By unifying the service search and composition problems into one composition-oriented service search problem, CoSA enables the dynamic and automatic service composition.

1 Introduction

When there do not exist the directly matched services but exist several services in the repository that can be combined to meet the requirements, how to find the composition relationship is the critical issue. We hope to be able to describe goals and leave it to the composition environment to figure out whether and how this can be implemented[1]. Composition-oriented service search is to find the composition relationships and the service implements from goal description. In essence composition oriented service search is a dynamic service composition problem. Dynamic service composition means two folds. One is to automatically discovery the simple service. The other is to automatically discovery the composition relationship among services.

The discovery methods for composition relationship can be classified into the manual method and the automatic one. But both manual and existing fully automated approaches have some problems[2]. With the increasing of the service categories and numbers, it is impossible for users to provide such composition relations. So the

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 530 – 536, 2006.

© Springer-Verlag Berlin Heidelberg 2006

automatic method is more feasible. But in the automatic method, the composition requesters usually only know the whole requirements instead of knowing which services will participate in the composition. Furthermore, many fully automated approaches usually make some unrealistic assumptions.

In this paper, we propose a composition-oriented and AND/OR graph-based service search method named as CoSA, which can automatically search the composition relation graph on concepts level and composition plan on implement level. We show that (1)an unified solution for search and composition, (2)automatic discovery of component services and their orchestration plans.

The rest of this paper is organized as follows. Section 2 gives the related concepts. Section 3 describes the composition-oriented service search algorithm CoSA, and gives an example. Section 4 discusses related works and gives the comparison with other similar methods. Finally, Section 5 draws some conclusions.

2 Related Concepts

The composition oriented service search can be divided into three sub problems:

1. How to describe the composition requirement?
One important part of composition requirement is the semantic description of the demanded service. Automatic service composition demands that the requirements are expressed in a quantitative and machine-readable format. Ontology technology is used in this paper. The goal or query is modeled as ontology. Ontology concepts and the relations between concepts are used to describe the service concepts and the composition relation concepts. All relevant facts such as available services, repositories are also expressed as ontologies.

2. How to describe and create the composition relation graph?
Composition relation graph describes the cooperative relationship among services. The composition operator is considered as AND node, and the candidate composition relationships as OR nodes. So the service composition can be transferred into a AND/OR graph search problem. The demanded composition relation graph is a solution graph of the AND/OR graph search algorithm.

3. How to create and evaluate the composition plan?
In order to get an executable application, each service concept in composition relation graph must be mapped to a concrete service implement. Thus the composition relation graph on concept level is transferred to the composition plan on the implement level.

To solve the three problems, this paper proposes a composition oriented service search algorithm CoSA that includes Composition And/Or Graph (CAOG) concept, CAOG-based composition relation generation algorithm and a composition plan generation algorithm.

Definition 1. Composition Relation Graph(*CRG*). CRG is a graph that describes the executing order, collaboration and interaction between services. CRG is expressed as an AND/OR graph, and has and only has a start node.

Definition 2. Composition plan (*CP*) is a path: *CP*= $v_1 - v_2 - \dots - v_n$, $v_i = (cr_i, s_i)$, $1 \leq i \leq n$, where $(cr_i, s_i) \in CR \times S$, *CR* is the composition request and S is the set of simple services or composite services. In order to simplify the problem, we consider the CP as an ordered set of services, that is: CP=$\{s_1, s_2, \dots, s_n\}$.

Definition 3. Service Semantic Description Model(*SSM)* is a seven tuple: *SSM* =<*UO*,*DO*,*OI*,*SI*,*α*,*β*,*γ*>, where *UO* is upper ontology. *DO* is domain ontology. *OI* is the ontology instance. *SI* is the service implementation. *A* means *DO* inherits from *UO*. *B* means *OI* is the instance of *DO*.γ means *SI* is the software implement of *OI*.

Definition 4. *UO* is a ten tuple[3]: *UO*=<*S*, *BA*, *COP*, *BR*,*ROP*,*P*, *φ*,*λ*,*η*, *rule*> where: *S*, *BA*, *COP*, *BR*, *ROP*, *P*, *rule* are all concepts for describing a service.

3 Composition-Oriented Service Search Algorithm-CoSA

The CoSA algorithm has following steps. Firstly, by defining the concept of *composition and/or graph* (CAOG) and transferring the reducing procedure from *SSM* to composition relation graph to the extending procedure of and/or graph, the composition relation graph is just gotten from the solution graph. Secondly, by using composability identifying rules and dynamic planning technology, the optimal composition plan is selected from solution graphs. Finally, a quantitative evaluation result for this optimal composition plan is given. Following is the algorithm description. Step 2 and 3 are the emphasis of this paper. Step 4 can refer to paper [4].

Algorithm 1. Composition oriented service search algorithm-CoSA
Input: goal ontology concept, constraints C, SSM model.
Output: composition plan P.

Steps:
1. p=null;
2. get the composition relation graph T by generate_CAOG() algorithm.
3. get optimal composition plan P by generate_CP() algorithm.
4. evaluate P.
5. return P.

3.1 Composition AND/OR Graph(CAOG)

Definition 5. Composition AND/OR Graph (CAOG) is a supergraph which is expressed as: CAOG=(V, E). V is the set of nodes and V is expressed as following:

$$V=\{v_i | v_i=(Concept, Num)\},\ 1 \leq i \leq n.$$

Where *Concept* means the service ontology concept in *SSM* model, *Num* means the number of service implements in the repository corresponding to service concept, *n* means the number of nodes.

E is the set of *k-linker*s. One *k-linker* means that a parent node points to a set of *k* successor-nodes. E is presented as following:

$$E=\{e_j | e_j=(f, v_k, \{v_i, v_{i+1}, \dots, v_{i+k-1}\})\},\ 1 \leq j \leq m,\ 1 \leq k \leq n$$

Where: $v_k \in V, v_i...v_{i+k-1} \in V$, and $v_k = f(v_i, v_{i+1}, ..., v_{i+k-1})$, *f* means the composition operator (namely, COP in definition 4) among $v_i, \ldots, v_{i+k-1}$ such as sequence, if-then-else, choose, loop and so on, which is similar to the process constructs in DAML-S[5]. *m* and n are the number of edges and nodes respectively.

In terms of whether there exist the implement instances corresponding to the S concept in *SSM*, we classify the *S* concepts into *abstract* and *stable* types. So the node type in CAOG can be classified into two types also as following.

Definition 6. *Abstract Type* (*AType*) means that there do not exist any implement instances in the repository corresponding to the service concept defined in *DO*.

Definition 7. *Stable Type* (*SType*) means that there exist implement instances in the repository corresponding to the service concept defined in *DO*.

Definition 8. *Target Node* (*TNode*). If a node belongs to *SType*, it is called *TNode*.

The key of heuristic search algorithm is to define a cost evaluation function.

Definition 9. Cost of *k*-linker. The cost of *k*-linker *op* can be calculated as: $h_c(op) = \alpha * weight + \beta * k$, where op is the name of the ontology concept corresponding to the *k-linker*'s parent node, α and β are two control parameters tuned by user. *k* is the number of successor-nodes of the linker. *Weight* is a value relevant to the composition operator. The weight-allocating method proposed in [6] is used. The weights of the composition operator such as *Sequence, If then else, Choose, Loop, Embeded and AndParallel* are 1,2,3,3,2,4 respectively.

Definition 10. For the service concept *s* in SSM that is *stable type*, the cost of their corresponding CAOG node is defined as: $h_b(s_{SType})=0$

Definition 11. Assume the service concept *s* in SSM is *abstract type*. In terms of the definition of *SSM* and *CAOG*, the following hold:

1) s_{AType}.composedBy()=cop, *cop* that is a composition operator;

2) cop.components()={$s_1, s_2, \ldots, s_k$}, s_i is the services of which s_{AType} consists .

3) *cop* is related to a *k*-linker:(cop, s_{AType},($s_1, s_2, \ldots, s_k$)). The cost of s_{AType}'s corresponding *CAOG* node is defined as: $h_b(s_{AType}) = h_c(\text{cop}) + h_b(s_1) + h_b(sc_2) + \ldots + h_b(s_k)$, where h_c(cop) can be calculated by definition 9. $s_1, s_2, \ldots, s_k$ can be of *abstract* or *stable* type.

Definition 12. The Cost of Solution Graph is h(*n*,*N*), where:

1)If *n* is one element in *N*, then h(n,N)=0.

2)If *n* has a outer linker which points to the successor-nodes {$n_1, \ldots, n_k$}, and assumes that the cost of the outer linker is $h_n(n)$, then $h(n,N) = h_b(n) + h(n_1,N) + \ldots + h(n_k,N)$

3.2 Generation of Composition Relation Graph

The principle of the generation algorithm for composition relation graph is that, the SSM model is considered as the search space, and all available composition relation among services is presented as Composition AND/OR Graph (CAOG). The user locates one service concept in SSM that corresponds to the root node of CAOG. In terms of the semantic relation between concepts in SSM, the search algorithm yields the middle nodes in CAOG. The service concept can include a group of sub service concepts. If the sub-concept belongs to *stable* type, it is the terminated node. The AO* algorithm is adopted to generate the composition relation graph.

3.3 Generation of Composition Plan

The composition relation graph (CRG) is on the concept level. In order to get the final executable application, it is necessary to select the optimal implement instances for each concept in CRG is another critical phase. This section we will propose a CP generating algorithm based on dynamic programming.

Algorithm 2. Composition Plan Generating Algorithm-generateCP()

Input: *CRG*, constraints *SC* on services, constraints *RC* on composition relations.
Output: composition plan - CP.
Steps:

1. For each concept in CRG, select the candidate service implements.
2. In terms of SC and RC, filter and validate the candidate service implements.
3. Preprocess the CRG in order that it is fitful to dynamic programming algorithm.
4. Transfer the problem of generating CP to an optimal problem, and make use of the dynamic programming algorithm to generate CP.

The basis of the algorithm is that the generation of CP is regarded as a multi-phase decision problem. Each phase decision problem identifies whether current service implements are able to compose with its proceeding node, and drop those unable to be composed together with current node. Through such filtering, the calculation complexity can be decreased greatly.

3.4 Example

This section takes a book shopping in web as an example to illustrate the CoSA algorithm. Figure 2 depicts the part of SSM model about book shopping. n_0 means the beginning node, and n_1, n_4, n_5, n_6, n_8, n_9, n_{10} are solution nodes.

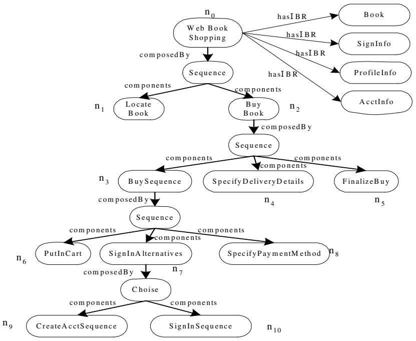
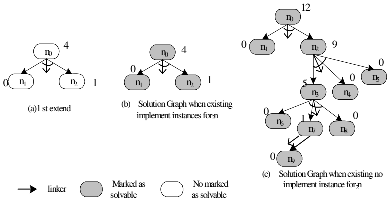


Fig. 1. Part of SSM Model about Services on Web Book-Shopping



(a)1 st extend

(b) Solution Graph when existing implement instances for $_2$n

(c) Solution Graph when existing no implement instance for $_2$n

Fig. 2. CAOG Search Graph

User proposes a query request as such concept description “Web Book Shopping”. Then the n_0 node will be found in SSM. The search procedures for composition relation are as depicted in figure 3. Assumed that there exist implement instances in repository for those services concepts that are n_1, n_4, n_5, n_6, n_8, n_9, n_{10}, the search procedure when the n_2 concept is of *abstract* or *stable* type is discussed as following.

1. When n_2 are of *stable* type. In terms of the extending algorithm, n_1 and n_2 will be extended from n_0 as depicted in figure 3(b). Because n_1 and n_2 are all solvable, n_0 is solvable also. The search is terminated.
2. When n_2 are of *abstract* type. the solution graph is as figure 3(c).

3.5 Algorithm Analysis

Based on SSM, the problem scale of CoSA algorithm is decreased greatly from the large number *R* of service implements in the repository to the number *N* of service concepts in SSM. *R* is usually much larger than *N*. In the CAOG generation algorithm, the composition structure and composition granularity are taken as the heuristic information to guide the search. This decreases the search space further. As for the composition plan-generating algorithm, the measurement for problem scope has two: length *L* of composition path and the number *M* of service implements for each service concepts in composition path. The time complexity of composition plan generating algorithm is $O(M^3*L)$ and the space cost is less than O(L).

4 Related Works

B.Arpinar proposed an ontology-driven service composition algorithm named as IMA[7]. The path calculation in IMA is based on each input and output. Thus the algorithm complexity will increase rapidly also. Our research just identify whether two services can be combined or not, the number of paths needed to check in CoSA is much little than IMA. M.Agarwal etal. proposed *Accord* algorighm[8] which demands that the relationship among services can be induced until all involved services have been selected out in advance. On contrast, our CoSA learns the composition relation from SSM. Kekta Fujii etal proposed the SeGSeC algorithm[9] in which the generation of composition path is prior to the semantic matching between services. However, in CoSA, the composition path is achieved by looking for the semantic model. QH Liang etal presents service search algorithm based on AND/OR graph[2]. Though both of our researches make use of AND/OR graph, there are differences between our methods. Firstly, the node definition in AND/OR graph is different. Secondly, SDG is defined by input and output of services in Liang’s method, while we use business semantic to described service dependencies. Thirdly, Liang’s method use bottom-up search algorithm while we use top-down search algorithm. Finally, the SDG in the former method is created dynamically after the user proposes a request. In CoSA, the SSM model exists prior to the request and evolves with the using history.

5 Conclusion and Future Works

This paper proposes the composition-oriented service search algorithm CoSA which is based on the Composition AND/OR Graph concept and dynamic programming thchnology.. Using the semantic support of SSM, the composition plan can be created automatically. CoSA features in the following. Firstly, ontology model is used to abstract the business concept and relations between the services in repository. These would cut down the search space greatly. Secondly, by transferring the service search problem to an optimal search problem, defining heuristic function in terms of composition operator and domain ontology and defining the concept of composition AND/OR graph, the search effectiveness can be improved. Thirdly, the CoSA method unifies the service search and composition problem into one composition-oriented service search problem instead of just focusing on one aspect. This leads us to further study the automatic service composition better. And it also provides supports for the dynamic and automatic service composition.

It is still necessary to conduct further empirical evaluation of the proposed method to investigate its adaptability that is not investigated in this paper. In addition, it is needed to add more heuristic information about the service trust evaluation. These await further researches.

Acknowledgments. This work is supported by the Fundamental Research Council of Harbin Engineering University in China.

References

1. Martin Hepp,Frank Leymann, John Domingue, Alexander Wahler,and Dieter Fensel Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management. Proceedings of the 2005 IEEE International Conference on e-Business Engineering (ICEBE'05)
2. Qianhui Althea Liang, Stanley Y W Su. AND/OR Graph and Search Algorithm for Discovering Composite Web Services. International Journal of Web Services Research. Hershey: Oct-Dec 2005. Vol. 2, Iss. 4; p.48-68
3. Xie Xiaoqin, Chen Kaiyun. Uniform Service Description With Semantics for Search and Composition. The International Multi-Symposiums on Computer and Computational Sciences (IMSCCS|06) June 12-16, 2006, IEEE Computer Press.
4. Xie Xiaoqin, Chen Kaiyun. An AHP-Based Evaluation Model for Service Composition. The 2006 International Conference on Computational Science and its Applications (ICCSA) 2006. Springer-Verlag Lecture Notes in Computer Science.
5. A.Ankolenkar et al. DAML-S: Web Service Description for the Semantic Web. Proc. 1st International Semantic Web Conference(ISWC), Springer Verlag, New York, 2002.348~363
6. Yingxu Wang. Component-Based Software Measurement. 2003. 247~262
7. B. Arpinar, R. Zhang, B. Aleman-Meza, and A. Maduko. Ontology-Driven Web Services Composition Platform. Journal of Information Systems and e-Business Management, Special issue on Service oriented enterprise IT applications and web services, 2004. IEEE International Conference on E-Commerce Technology (CEC'04) July 06 - 09, 2004 San Diego, California. 2004. 146~152
8. Manish Agarwal; Manish Parashar. Enabling autonomic compositions in grid environments. Grid Computing,2003. Proceedings. Fourth International Workshop on 17 Nov. 2003.34~41
9. K. Fujii and T. Suda. Dynamic Service Composition Using Semantic Information. the 2nd International Conference on Service Oriented Computing (ICSOC '04), November 2004.

DODDLE-OWL: A Domain Ontology Construction Tool with OWL

Takeshi Morita[1], Naoki Fukuta[2], Noriaki Izumi[3], and Takahira Yamaguchi[1]

[1] Keio University, 4-1-1 Hiyoshi, Kohokuku, Yokohama-shi, 223-8522 Japan
{t_morita, yamaguti}@ae.keio.ac.jp
[2] Shizuoka University, 3-5-1 Johoku, Hamamatsu, Shizuoka 432-8011 Japan
fukuta@cs.inf.shizuoka.ac.jp
[3] National Institute of AIST, 1-18-13 Sotokanda, Chiyoda-ku,Tokyo 101-0021 Japan
n.izumi@aist.go.jp

Abstract. In this paper, we propose a domain ontology construction tool with OWL. The advantage of our tool is focusing the quality refinement phase of ontology construction. Through interactive support for refining the initial ontology, OWL-Lite level ontology, which consists of taxonomic relationships (defined as classes) and non-taxonomic relationships (defined as properties), is constructed effectively. The tool also provides semi-automatic generation of the initial ontology using domain specific documents and general ontologies.

1 Introduction

The Semantic Web [1] is now gathering attentions from researchers in wide area. Adding semantics (meta-data) to the Web contents, software agents are able to understand and even infer Web resources. To realize such paradigm, the role of ontologies [2] is important in terms of sharing common understanding among both people and software agents [3]. In knowledge engineering field ontologies have been developed for particular knowledge system mainly to reuse domain knowledge. On the other hand, for the Semantic Web, ontologies are constructed in distributed places or domain, and then mapped each other. For this purpose, it is an important task to realize a software environment for rapid construction of ontologies for each domain. Towards the on-the-fly ontology construction, many researches are focusing on automatic ontology construction from existing Web resources, such as dictionaries, by machine processing with concept extraction algorithms. However, depending on domains (a law domain etc.), the important concepts which doesn't occur frequently in the resources may be required to be added by hand for ontology construction. In such a domain, if a user doesn't intervene, constructing ontologies cannot readily be done. Considering such situation, we believe that the most important aspect of the on-the-fly ontology construction is that how efficiently the user is able to complete making the ontology for the Semantic Web contents available to the public. For this reason, ontologies should be constructed not fully automatically, but through interactive support by software environment from the early stage of ontology construction.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 537–551, 2006.
© Springer-Verlag Berlin Heidelberg 2006

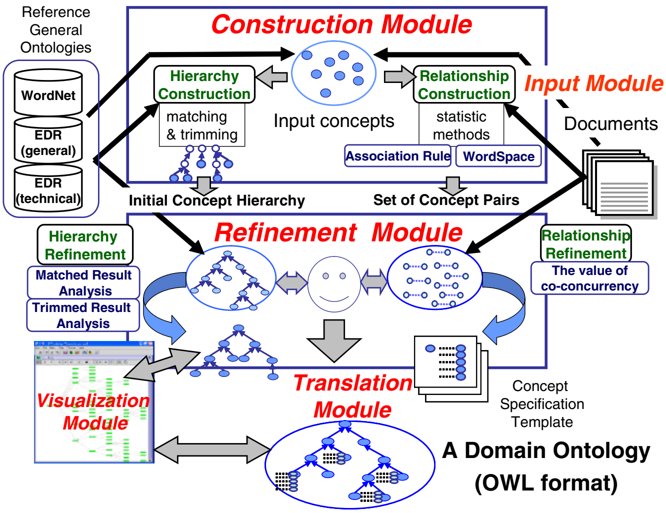


Fig. 1. DODDLE-OWL overview

Although it may seem to be contradiction in terms of efficiency, the total cost of ontology construction would become less than automatic construction since if the ontology is constructed with careful interaction between the system and the user, less miss-construction will be happened. It also means that high-quality ontology would be constructed.

In this paper, we propose a domain ontology construction tool with OWL named DODDLE-OWL (a Domain Ontology rapiD DeveLopment Environment - OWL [4] extension). The architecture of DODDLE-OWL is re-designed based on DODDLE-II [5], the former version of DODDLE-OWL. DODDLE-OWL has the following five modules: Input Module, Construction Module, Refinement Module, Visualization Module, and Translation Module. Especially, to realize the user-centered environment, DODDLE-OWL dedicates to Refinement Module. It enables us to develop ontologies with interactive indication of which part of ontology should be refined. DODDLE-OWL supports the construction of both taxonomic relationships and non-taxonomic relationships in ontologies. Since DODDLE-II has been built for ontology construction not for the Semantic Web but for typical knowledge systems, it needs some extensions for the Semantic Web such as OWL (Web Ontology Language) [4] export facility. DODDLE-OWL contributes the evolution of ontology construction and the Semantic Web.

2 DODDLE-OWL Architecture

Figure 1 shows the overview of DODDLE-OWL. DODDLE-OWL has following five modules: Input Module, Construction Module, Refinement Module, Visualization Module, and Translation Module. Here, we assume that there are one

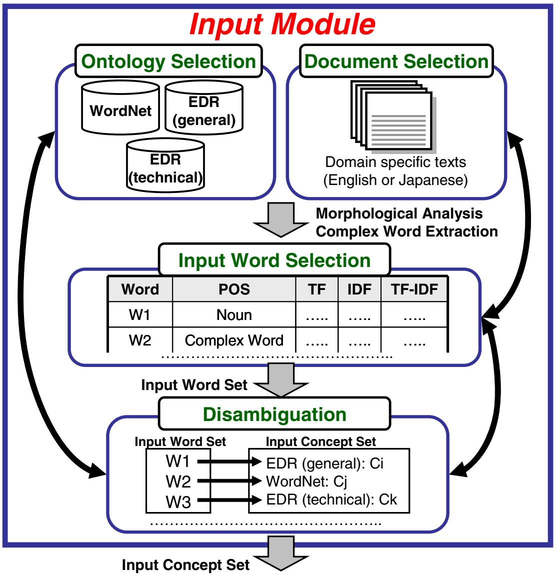


Fig. 2. Input Module

or more domain specific documents, and we also assume that the user can select important words that are needed to construct a domain ontology. First, as input of DODDLE-OWL, a user selects several concepts in Input Module. The detail of Input Module is described in Section 2.1. In Construction Module, DODDLE-OWL generates the basis of an ontology, an initial concept hierarchy and set of concept pairs, by referring to general ontologies and documents. The detail of Construction Module is described in Section 2.2. In Refinement Module, the initial ontology generated by Construction Module is refined by the user through interactive support by DODDLE-OWL. The detail of Refinement Module is described in Section 2.3. The ontology constructed by DODDLE-OWL can be exported with the representation of OWL. Finally, Visualization Module (MR^3 [6]) is connected with DODDLE-OWL and works with an graphical editor.

2.1 Input Module

Figure 2 shows the procedure of Input Module. Input Module consists of Ontology Selection Module, Document Selection Module, Input Word Selection Module, and Disambiguation Module. Input concepts that are significant concepts in the domain are selected in Input Module.

First, a user selects general ontologies in Ontology Selection Module. DODDLE-OWL can refer WordNet [7], EDR [8] general vocabulary dictionary, and EDR technical terminology dictionary (Information Processing) as general ontologies to construct classes and properties. The user can select some general ontologies from among those general ontologies.

Second, in Document Selection Module, the user selects domain specific documents described in English or Japanese. At this phase, the user can select to extract words of what part of speech (POS).

Third, in Input Word Selection Module, DODDLE-OWL shows a list of extracted words including complex words, POS, TF (Term Frequency), IDF (Inverse Document Frequency), TF-IDF in the documents. Domain specific documents contain many significant complex words. Therefore, extracting complex words is needed to construct domain ontologies. At this phase, while considering POS, TF, and so on, the user selects input words that are significant words for the domain.

Finally, in Disambiguation Module, the user identifies the sense of input words to map those words to concepts in the general ontologies. A word has many senses. Therefore, there are many concepts that correspond to a word. Disambiguation Module shows input words and concepts that correspond to the input words. While considering the domain, the user selects most appropriate concept for a word.

The headwords of most concepts do not contain complex words. Therefore, disambiguation of complex words is difficult. Disambiguation module uses `partial match` to disambiguate most of complex words. Disambiguation Module uses `perfect match` and `partial match` to disambiguate input words. `Perfect match` means an input word corresponds to a headword of a concept perfectly. `Partial match` means an input word corresponds to a headword of a concept partially. The priority of `perfect match` is higher than that of `partial match`. If an input word does not correspond perfectly to any headword of concepts in the general ontologies, Disambiguation Module analyzes the morphemes of the input word (especially in Japanese). The intput word can be considered a list of the morpheme. Disambiguation Module tries to correspond the sublist containing the last morpheme of the list to the concepts of the general ontologies. The input word is corresponded to the concepts which have the longest sublist.

For example, “rocket delivery system” does not correspond to the headwords of concepts in the general ontologies perfectly. Disambiguation Module analyzes morphemes of “rocket delivery system”. “Rocket delivery system” is resolved to “rocket”, “delivery”, and “system”. First, Disambiguation Module disambiguates ”delivery system“. Then, Disambiguation Module disambiguates ”system“. In this example, ”delivery system“ does not correspond to the headwords of concepts in the general ontologies. On the other hand, ”system“ corresponds to the headwords of concepts in the general ontologies. Consequently, Disambiguation Module shows the concepts that have ”system“ as their headword to disambiguate “rocket delivery system”.

Input words which do not correspond to the headwords of concepts in the general ontologies are `undefined words`. If appropriate concepts do not exist in the general ontologies, the input words are also undefined words. The user defines the undefined words manually in Refinement Module.

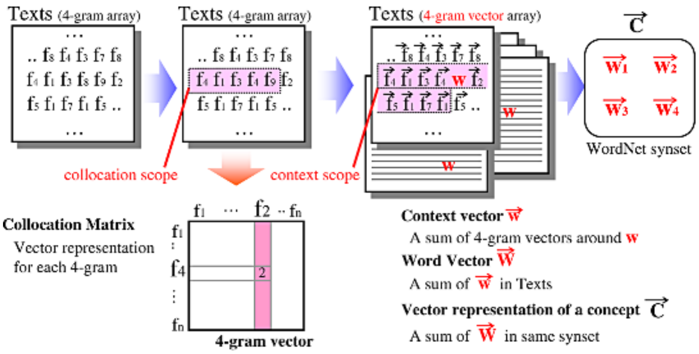


Fig. 3. Construction flow of WordSpace

2.2 Construction Module

In Construction Module, DODDLE-OWL generates the basis of output ontology for further modification by a user. The upper side of Figure 1 describes the procedure of Construction Module. Construction Module consists of two sub-modules: Hierarchy Construction Module and Relationship Construction Module.

For building taxonomic relationship of an ontology, Hierarchy Construction Module attempts to extract `best-matched concepts` that are concepts matching between input concepts and general ontologies' concepts perfectly. Matched nodes are extracted, and merged at each root nodes. This is called `initial model`. The initial model has unnecessary internal nodes. They do not contribute in keeping topological relationships among matched nodes, such as parent-child relationship and sibling relationship. Therefore, we get a `trimmed model` by trimming the unnecessary internal nodes from the initial model. The partial matched words are defined as sub-concept of the concepts which are selected in Disambiguation Module. Then, following [9] partial matched words are reconstructed. An initial concept hierarchy is constructed as an IS-A hierarchy.

To extract related concept pairs from domain specific documents as a basis of identifying non-taxonomic relationships, co-occurrency based statistic methods are applied. In particular, WordSpace [10] and an association rule algorithm [11] are used in Relationship Construction Module. These extracted pairs are considered to be closely related and that will be used as candidates to refine and add non-taxonomic relations. In Refinement Module, the user identifies some relationship between concepts in the pairs. The detail of WordSpace and an association rule algorithm are described as follows.

Construction of WordSpace. WordSpace is constructed as shown in Figure 3.

1. *Extraction of high-frequency 4-grams*
   Since letter-by-letter co-occurrence information becomes too much and so often irrelevant, we take term-by-term co-occurrence information in four words

(4-gram) as the primitive to make up co-occurrence matrix useful to represent context of a text based on experimented results. We take high frequency 4-grams in order to make up WordSpace.

2. *Construction of collocation matrix*
   A *collocation matrix* is constructed in order to compare the context of two 4-grams. Element $a_{i,j}$ in this matrix is the number of 4-gram f_i which comes up just before 4-gram f_j (called *collocation area*). The collocation matrix counts how many other 4-grams appear before the target 4-gram. Each column of this matrix is the *4-gram vector* of the 4-gram f.
3. *Construction of context vectors*
   A *context vector* represents context of a word or phrase in a text. A sum of 4-gram vectors around appearance place of a word or phrase (called *context area*) is a context vector of a word or phrase in the place.
4. *Construction of word vectors*
   A word vector is a sum of context vectors at all appearance places of a word or phrase within texts, and can be expressed with Eq.1. Here, $\tau(w)$ is a vector representation of a word or phrase w, $C(w)$ is appearance places of a word or phrase w in a text, and $\varphi(f)$ is a 4-gram vector of a 4-gram f. A set of vector $\tau(w)$ is WordSpace.

$$\tau(w) = \sum_{i \in C(w)} \Big(\sum_{f \text{ close to } i} \varphi(f) \Big) \tag{1}$$

5. *Construction of vector representations of all concepts*
   The best matched "synset" of each input words in WordNet is already specified, and a sum of the word vector contained in these synsets is set to the vector representation of a concept corresponding to an input term. The concept label is the input term.
6. *Construction of a set of similar concept pairs*
   Vector representations of all concepts are obtained by constructing WordSpace. Similarity between concepts is obtained from inner products in all the combination of these vectors. Then we define certain threshold for this similarity. A concept pair with similarity beyond the threshold is extracted as a similar concept pair.

Finding Association Rules between Input Words. The basic association rule algorithm is provided with a set of transactions, $T := \{t_i \mid i = 1..n\}$,where each transaction t_i consists of a set of items, $t_i = \{a_{i,j} \mid j = 1..m_i, a_{i,j} \in C\}$ and each item $a_{i,j}$ is a set of concepts C. The algorithm finds association rules $X_k \Rightarrow Y_k : (X_k, Y_k \subset C, X_k \cap Y_k = \{\})$ such that measures for support and confidence exceed user-defined thresholds. Thereby, support of a rule $X_k \Rightarrow Y_k$ is the percentage of transactions that contain $X_k \cup Y_k$ as a subset (Eq.2) and confidence for the rule is defined as the percentage of transactions that Y_k is seen when X_k appears in a transaction (Eq.3).

$$support(X_k \Rightarrow Y_k) = \frac{\mid \{t_i \mid X_k \cup Y_k \subseteq t_i\} \mid}{n} \tag{2}$$

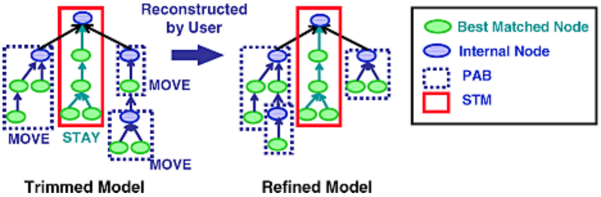


Fig. 4. Matched Result Analysis

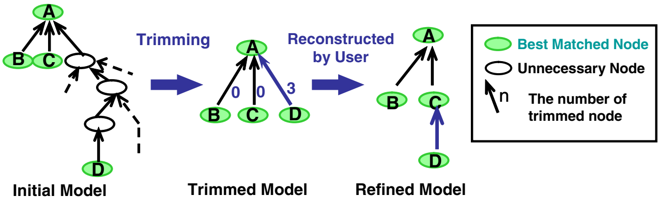


Fig. 5. Trimmed Result Analysis

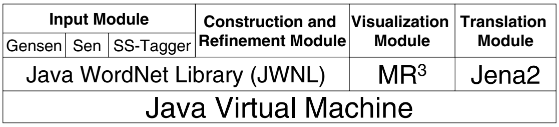
$$confidence(X_k \Rightarrow Y_k) = \frac{\mid \{t_i \mid X_k \cup Y_k \subseteq t_i\} \mid}{\mid \{t_i \mid X_k \subseteq t_i\} \mid} \qquad (3)$$

As we regard input words as items and sentences in documents as transactions, DODDLE-OWL finds associations between words in the documents. Based on experimented results, we define the threshold of support as 0.4% and the threshold of confidence as 80%. When an association rule between words exceeds both thresholds, the pair of words is extracted as candidates for non-taxonomic relationships.

2.3 Refinement Module

In order to refine the initial ontology, Refinement Module manages concept drift and evaluates set of concept pairs interactively with a user. The lower side of Figure 1 shows the procedure of Refinement Module. Since the initial taxonomy is constructed from general ontologies, we need to adjust the taxonomy to the specific domain considering an issue called concept drift. It means that the position of particular concepts changes depending on the domain. For concept drift management, DODDLE-OWL applies two strategies: Matched Result Analysis (Figure 4) and Trimmed Result Analysis (Figure 5).

In Matched Result Analysis, DODDLE-OWL divides the taxonomy into PABs (PAths including only Best matched concepts) and STMs (SubTrees that includes best-matched concepts and other concepts and so can be Moved) and indicates on the screen. PABs are paths that include only best-matched concepts that have senses suitable for the given domain. STMs are subtrees of which root is an internal concept of WordNet and its subordinates are all best-matched concepts. Since the sense of an internal concept has not been identified by a user yet,



<table>
<tr><th colspan="3">Input Module</th><th rowspan="2">Construction and Refinement Module</th><th rowspan="2">Visualization Module</th><th rowspan="2">Translation Module</th></tr>
<tr><th>Gensen</th><th>Sen</th><th>SS-Tagger</th></tr>
<tr><td colspan="4">Java WordNet Library (JWNL)</td><td>MR^3</td><td>Jena2</td></tr>
<tr><td colspan="6">Java Virtual Machine</td></tr>
</table>

Fig. 6. DODDLE-OWL Implementation Architecture

STMs may be moved to other places for the concept adjustment to the domain. In addition, for Trimmed Result Analysis, DODDLE-OWL counts the number of internal concepts when the part was trimmed. By considering this number as the original distance between those two concepts, DODDLE-OWL indicates to move the lower concept to other places.

As a facility for related concept pair discovery, there are functions that allow users to attempt some ways to improve the quality of extracted concept pairs through trial and error by changing parameters of statistic methods. Users can re-adjust the parameters of WordSpace and association rule algorithm and check the result. After that, DODDLE-OWL generates `Concept Specification Templates` by using the results. It consists of some concept pairs which have considerable relationship found from the result value of statistic methods. By referring to the constructed domain specific taxonomic relationship and the `Concept Specification Templates`, a user constructs a domain ontology.

3 Implementation

Figure 6 shows DODDLE-OWL implementation architecture. DODDLE-OWL is implemented in Java language. Input Module, Construction Module, and Refinement Module use Java WordNet Library (JWNL) [12] to access WordNet. Input Module uses Gensen [13], Sen [14], and SS-Tagger [15]. Gensen is used for extracting japanese and english complex words. Sen is a Japanese morphological analyzer implemented in Java language. Sen is used for extracting Japanese words and its POS from documents. SS-Tagger is an English POS tagger. SS-Tagger is used for extracting English words and its POS from documents. MR^3 [6] is used as Visualization Module. MR^3 is an RDF(S) graphical editor with meta-model management facility such as consistency checking of classes and a model in which these classes are used as the type of instances. Translation Module uses Jena2 Semantic Web Framework [16] to export a constructed ontology in OWL format.

Figure 7 shows a typical usage of DODDLE-OWL. DODDLE-OWL's user interface consists of Ontology Selection Panel, Document Selection Panel, Inpu Word Selection Panel, Disambiguation Panel, Construction and Refinement Panel for Classes, Construction and Refinement Panel for Properties, Visualization Module, and Construction and Refinement Panel for Relationships. First, the user selects general ontologies in Ontology Selection Panel ((1) in Figure 7). Second, in Document Selection Panel ((2) in Figure 7), the user opens domain specific documents described in Japanese or English. In this panel, words in the documents

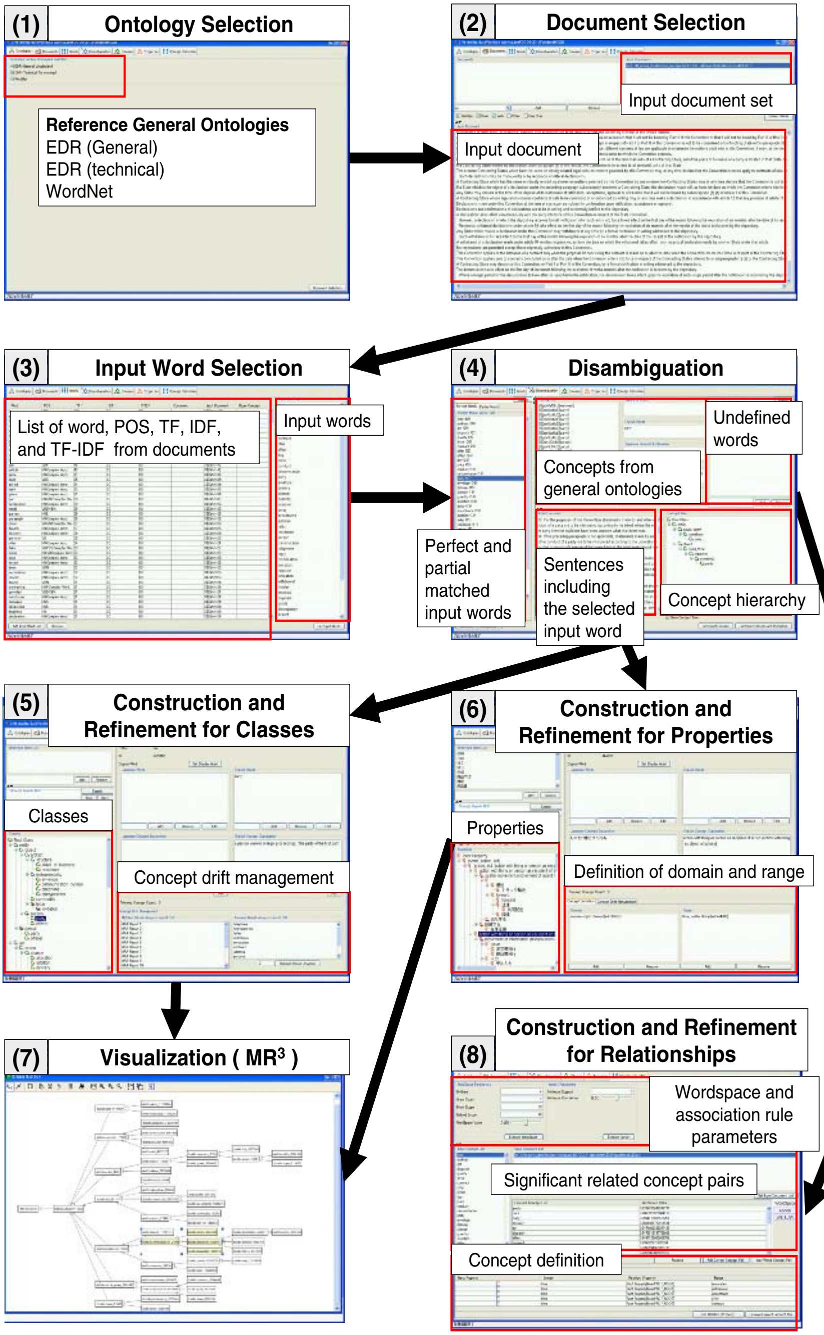

(1) Ontology Selection
Reference General Ontologies
EDR (General)
EDR (technical)
WordNet
(2) Document Selection
Input document set
Input document
(3) Input Word Selection
List of word, POS, TF, IDF, and TF-IDF from documents
Input words
(4) Disambiguation
Undefined words
Concepts from general ontologies
Perfect and partial matched input words
Sentences including the selected input word
Concept hierarchy
(5) Construction and Refinement for Classes
Classes
Concept drift management
(6) Construction and Refinement for Properties
Properties
Definition of domain and range
(7) Visualization (MR³)
(8) Construction and Refinement for Relationships
Wordspace and association rule parameters
Significant related concept pairs
Concept definition

Fig. 7. A typical usage of DODDLE-OWL

are extracted. Third, in Input Word Selection Panel ((3) in Figure 7), the user selects input words that are significant words for the domain. The user can sort the extracted words based on POS, TF, IDF, and TF-IDF in this Panel. Fourth, in Disambiguation Panel ((4) in Figure 7), the user associates the input words with concepts by referring the general ontologies which are selected in Ontology Selection Panel. After mapping input words and corresponding concepts, an initial class and property hierarchy are generated. Also set of concept pairs are extracted by co-occurrency based statistic methods such as WordSpace method and the association rule learner by default parameters. (5) and (6) of Figure 7 shows the Construction and Refinement Panel for Classes and Properties. These panels indicate some groups of concepts in the taxonomy so that the user can decide which part should be refined. (7) of Figure 7 shows the display of concept drift management in Visualization Module. (8) of Figure 7 shows Construction and Refinement Panel for Relationships. This panel is used for setting parameters used in the WordSpace method and the association rule algorithm to apply to documents in order to generate significantly related concept pairs. In WordSpace method, there are parameters such as the gram number (default gram number is four), minimum N-gram count (to extract high-frequency grams only), front scope and behind scope in the text. In the association rule learner, minimum confidence and minimum support are set by the user. Finally, the user can export a constructed domain ontology in OWL format.

4 Case Studies

In order to evaluate how DODDLE-OWL is doing in a practical field, case studies have been done in particular field of business. The particular field of business is called “XML Common Business Library” (xCBL) [17]. 57 business words are extracted by a user from xCBL Document Reference (about 2,500 words). The user is not an expert but has business knowledge.

4.1 Taxonomic Relationship Acquisition

Table 1 shows the number of concepts in each model under taxonomic relationship acquisition and table 2 shows the evaluation of two strategies by the user. The recall per subtree is more than 0.5 and is good. The precision and the recall per path are less than 0.3 and are not so good, but about 80 % portions of taxonomic relationships were constructed with Hierarchy Construction Module and Hierarchy Refinement Module support.

4.2 Non-taxonomic Relationship Learning

Relationship Construction. High-frequency 4-grams (sets of four words that co-occur term-by-term) were extracted from xCBL Document Reference standard form conversion removed duplication, and 1,240 kinds of 4-grams were obtained. In order to keep density of a collocation matrix high, the extraction frequency of 4-grams must be adjusted according to the scale of documents. In

Table 1. The Change of the Number of Concepts under Taxonomic Relationship Acquisision

Model	Input Words	Initial Model	Triimed Model	Concept Hierarchy
# Concept	57	152	83	82

Table 2. Precision and Recall in the Case Study with xCBL

	Precision	Recall per Path	Recall per Subtree
Matched Result	0.2 (5/25)	0.29 (5/17)	0.71 (5/7)
Trimmed Result	0.22 (2/9)	0.13 (2/15)	0.5 (2/4)

Table 3. Evaluation by the User with xCBL definition

	WordSPace (WS)	Association Rules (AR)	The Join of WS and AR
# Extracted concept pairs	40	39	66
# Accepted concept pairs	30	20	39
# Rejected concept pairs	10	19	27
Precision	0.75 (30/40)	0.51 (20/30)	0.59 (39/66)

order to construct a context vector, a sum of 4-gram vectors around appearance place circumference of each of 57 concepts was calculated. In order to construct a context scope from some 4-grams, it consists of putting together 10 4-grams before the 4-gram and 10 4-grams after the 4-grams independently of length of a sentence. For each of 57 concepts, the sum of context vectors in all the appearance places of the concept in xCBL was calculated, and the vector representations of the concepts were obtained. The set of these vectors is used as WordSpace to extract concept pairs with context similarity. Having calculated the similarity from the inner product for concept pairs which is all the combination of 57 concepts, 40 concept pairs were extracted.

DODDLE-OWL extracted 39 pairs of words from the document using the association rule algorithm. There are 13 pairs out of them in a set of similar concept pairs extracted using WordSpace. Then, DODDLE-OWL constructed `Concept Specification Templates` from two sets of concept pairs extracted by WordSpace and association rule algorithm.

Evaluation of Results of Relationship Refinement. The user evaluated the following two sets of concept pairs: one is extracted by WS (WordSpace) and the other is extracted by AR (Association Rule algorithm). Figure 8 shows two different sets of concept pairs from WS and AR. It also shows portion of extracted concept pairs that were accepted by the user. Table 3 shows the details of evaluation by the user, computing precision only. Since the user didn't define concept definition in advance, we can not compute recall. Looking at the field

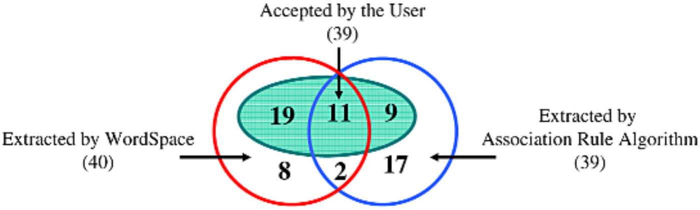


Fig. 8. Two Difference Sets of Concept Pairs from WS and AR and Concept Sets have Relationships

of precision in Table 3, the precision from WS is higher than others. Most of concept pairs which have relationships were extracted by WS. The percentage is about 77% (30/39). But there are some concept pairs which were not extracted by WS. Therefore, taking the join of WS and AR is the best method to support a user to construct non-taxonomic relationships.

4.3 Results and Evaluation of Case Studies

In regards to support in constructing taxonomic relationships, the precision and recall are less than 0.3 in the case study. 80 % or more support comes from Hierarchy Construction Module and Hierarchy Refinement Module. About more than half portion of the final domain ontology results in the information extracted from WordNet. Since the two strategies just imply the part where concept drift may come up, the part generated by them has low component rates and about 30 % hit rates. So one out of three indications based on the two strategies work well in order to manage concept drift. The two strategies use matched and trimmed results, therefore based on structural information of an MRD only, the hit rates are not so bad. In order to manage concept drift smartly, we may need to use more semantic information that is not easy to come up in advance in the strategies, and we also may need to use domain specific documents and other information resource to improve supporting a user in constructing taxonomic relationships.

In regards to construction of non-taxonomic relationships, the precision in the case study with xCBL is good. Generating non-taxonomic relationships of concepts is harder than modifying and deleting them. Therefore, DODDLE-OWL supports the user in constructing non-taxonomic relationships.

After analyzing results of case studies, we have the following problems.

1. *Determination of a Threshold*
   Threshold of the context similarity changes in effective value with each domain. It is hard to set up the most effective value in advance.
2. *Specification of a Concept Relation*
   `Concept Specification Templates` have only concept pairs based on the context similarity, it requires still high cost to specify relationships between them. It is needed to support specification of concept relationships on this system in the future work.

3. *Ambiguity of Multiple Terminology*
   For example, the term "transmission" is used in two meanings, "transmission (of goods)" and "transmission (of communication)", in a document, but DODDLE-OWL considers these words as the same and creates WordSpace as it is. Therefore constructed vector expression may not be exact. In order to extract more useful concept pairs, semantic specialization of a multisense word is necessary, and it should be considered that the 4-grams with same appearance and different meaning are different 4-grams.

5 Related Work

Navigli et,al. proposed OntoLearn [18], that supports domain ontology construction by using existing ontologies and natural language processing techniques. In their approach, existing concepts from WordNet are enriched and pruned to fit the domain concepts by using NLP (Natural Language Processing) techniques. They argue that the automatically constructed ontologies are practically usable in the case study of a terminology translation application. However, they did not show any evaluations of the generated ontologies themselves that might be done by domain experts. Although a lot of useful information is in the machine readable dictionaries and documents in the application domain, some essential concepts and knowledge are still in the minds of domain experts. We did not generate the ontologies themselves automatically, but suggests relevant alternatives to the human experts interactively while the experts' construction of domain ontologies. In another case study [19], we had an experience that even if the concepts are in the MRD (Machine Readable Dictionary), they are not sufficient to use. In the case study, some parts of hierarchical relations are counterchanged between the generic ontology (WordNet) and the domain ontology, which are called "Concept Drift". In that case, presenting automatically generated ontology that contains concept drifts may cause confusion of domain experts. We argue that the initiative should be kept not on the machine, but on the hand of the domain experts at the domain ontology construction phase. This is the difference between our approach and Navigli's. Our human-centered approach enabled us to cooperate with human experts tightly.

From the technological viewpoint, there are two different related research areas. In the research using verb-oriented method, the relation of a verb and nouns modified with it is described, and the concept definition is constructed from this information (e.g. [20]). In [21], taxonomic relationships and Subcategorization Frame of verbs (SF) are extracted from technical texts using a machine learning method. The nouns in two or more kinds of different SF with the same frame-name and slot-name are gathered as one concept, base class. And ontology with only taxonomic relationships is built by carrying out clustering of the base class further. Moreover, in parallel, Restriction of Selection (RS) which is slot-value in SF is also replaced with the concept with which it is satisfied instantiated SF. However, proper evaluation is not yet done. Since SF represents the syntactic relationships between verb and noun, the step for the conversion to non-taxonomic relationships is necessary.

On the other hand, in ontology learning using data-mining method, discovering non-taxonomic relationships using an association rule algorithm is proposed by [22]. They extract concept pairs based on the modification information between words selected with parsing, and made the concept pairs a transaction.

By using heuristics with shallow text processing, the generation of a transaction more reflects the syntax of texts. Moreover, RLA, which is their original learning accuracy of non-taxonomic relationships using the existing taxonomic relations, is proposed. The concept pair extraction method in our paper does not need parsing, and it can also run off context similarity between the words appeared apart each other in texts or not mediated by the same verb.

6 Conclusion

In this paper, we presented a domain ontology construction tool with OWL named DODDLE-OWL, which aims at a total support environment for user-centered on-the-fly ontology construction. Its main principle is that high-level support for users through interaction. First, in Input Module, the user selects significant concepts for the domain from general ontologies. Then, Construction Module generates the basis of an ontology in the forms of an initial concept hierarchy and set of concept pairs, by referring to general ontologies and domain specific documents. Refinement Module provides management facilities for concept drift in the taxonomy and identifying significant set of concept pairs in extracted related concept pairs. Finally, Translation Module exports the ontology with the representation of OWL.

According to case studies for constructing taxonomic relationships, 80 % or more support comes from Hierarchy Construction and Refinement Module. According to case studies for constructing non-taxonomic relationships, it is important to select combinations of algorithms to get better candidate of set of concept pairs.

At the moment, we have developed a rocket operation ontology using DODDLE-OWL. The ontology has been developed from approximately 2,000 Japanese documents and includes approximately 40,000 concepts. We'd like to evaluate the ontology and show the practical utility of DODDLE-OWL. As future work, we'd like to use existing domain ontologies using ontology search engine like Swoogle [23].

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001)
2. Heijst, G.V.: The Role of Ontologies in Knowledge Engineering. Dr.thesis, University of Amsterdam (1995)
3. Ding, Y., Foo, S.: Ontology Research and Development, Part 1 – a Review of Onlotogy. Journal of Information Science (2002) 123–136
4. Michael K. Smith, C.W., McGuinness, D.L.: OWL Web Ontology Language Guide (2004) http://www.w3.org/TR/owl-guide/.

5. Kurematsu, M., Iwade, T., Nakaya, N., Yamaguchi, T.: Doddle ii a domain ontology development environment using a mrd and text corpus. IEICE(E) **E87-D**(4) (2004) 908–916
6. Takeshi Morita and Noriaki Izumi and Naoki Fukuta and Takahira Yamaguchi: A graphical rdf-based meta-model management tool. IEICE(E) **E89-D**(4) (2006) 1368–1377
7. G.A.Miller: WordNet: A Lexical Database for English. ACM **38**(11) (1995) 39–41
8. National Institute of Information and Communications Technology: (EDR Electronic Dictionary Technical Guide) http://www2.nict.go.jp/kk/e416/EDR/ENG/E_TG/E_TG.html.
9. Velardi, P., Fabriani, P., Missikoff, M.: Using Text Processing Techniques to Automatically enrich a Domain Ontology. Proceedings of the international conference on Formal Ontology in Information Systems (2001) 270–284
10. Marti A. Hearst, H.S.: Customizing a Lexicon to Better Suit a Computational Task. Corpus Processing for Lexical Acquisition (1996) 77–96
11. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. Proceedings of VLDB Conference (1994) 487–499
12. : (Java wordnet library) http://jwordnet.sourceforge.net/.
13. Nakagawa, H., Mori, T.: A simple but powerful automatic term extraction method. Computerm2: 2nd International Workshop on Computational Terminology, COLING-2002 WORKSHOP (2002) 29–35
14. : (Sen) http://ultimania.org/sen/.
15. Tsuruoka, Y., Tsujii, J.: Bidirectional inference with the easiest-first strategy for tagging sequence data. Proceedings of HLT/EMNLP (2005) 467–474
16. HP Labs: Jena Semantic Web Framework. (2003) http://jena.sourceforge.net/downloads.html.
17. One, C.: (xcbl:xml common business library) http://www.xcbl.org/.
18. Navigli, R., Velardi, P.: Automatic adaptation of wordnet to domains. Proceedings of International Workshop on Ontologies and Lexical Knowledge Bases (2002)
19. Yamaguchi, T.: Constructing domain ontologies based on concept drift analysis. IJCAI Workshop on Ontologies and Problem-Solving Methods (1999) 13–1–13–7
20. Hahn, U., Schnattingerg, K.: Toward text knowledge engineering. AAAI-98 proceedings (1998) 524–531
21. Faure, D., Nedellec, C.: Knowledge Acquisition of Predicate Argument Structures from Technical Texts. Proceedings of International Conference on Knowledge Engineering and Knowledge Management (1999) 329–334
22. Maedche, A., Staab, S.: Discovering Conceptual Relations from Text. Proceedings of 14th European Conference on Artificial Intelligence (2000) 321–325
23. Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y., Kolari, P.: Finding and ranking knowledge on the semantic web. Proceedings of the 4th International Semantic Web Conference, LNCS 3729 (2005) 156–170 http://swoogle.umbc.edu/.

Knowledge Elicitation Plug-In for Protégé: Card Sorting and Laddering

Yimin Wang[1], York Sure[1], Robert Stevens[2], and Alan Rector[2]

[1] Institute AIFB, University of Karlsruhe, Germany
{ywa, sure}@aifb.uni-karlsruhe.de
[2] School of Computer Science, University of Manchester, M13 9PL, UK
{rector, robert.stevens}@cs.man.ac.uk

Abstract. Ontologies have been widely accepted as the primary method of representing knowledge in the Semantic Web. Knowledge Elicitation (KE) is usually one of the first steps in building ontologies. A number of ontology editors such as Protégé have been developed to assist users in building ontologies efficiently. However, traditional KE techniques, such as card sorting and laddering, are not yet supported, but performed manually and outside of such tools. In this paper we present a methodology and a corresponding plug-in for Protégé that allows graphical ellicitation knowledge from documents using card sorting and laddering approaches. Our aim is to seamlessly integrate the KE techniques into the ontology building process to make ontology building more efficient and less error-prone. As a side-effect the persistent storage of card sorting and laddering results allows for later traceability of ontology development. KE largely depends on user interaction with the plug-in, therefore we employed user-centred design principles to capture requirements. After implementation, the plug-in was evaluated thoroughly against the requirements. The evaluation shows that this KE plug-in meets many of the user's expectations and indeed saves them considerable time when building ontologies.

1 Introduction

The explosion of digital knowledge makes finding accurate information effectively an increasingly important topic. Making knowledge explicit, e.g. in the form of ontologies and corresponding metadata, offers many opportunities to facilitate effective knowledge access. One of the first steps in building ontologies is usually a knowledge elicitation (KE) process, which is also known as an important branch of knowledge acquisition. Traditional KE is a kind of labour-intensive manual work, extremely time-consuming, and often not well connected to further steps in ontology engineering. What is needed are more usable, handy and in particular well-integrated toolkits for knowledge elicitation.

Several standard knowledge acquisition/elicitation techniques, such as card sorting and laddering, have been developed to help in organising domain expert's ideas into basic structures and to recover tacit knowledge. Card sorting has been used for several decades, and it is remarkably useful for finding out how people categorise things [1,2]. Laddering was first introduced by Hinkle [3], a clinical psychologist, in order to model the concepts and beliefs of people by an unambiguous and systematic approach. Most

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 552–565, 2006.
© Springer-Verlag Berlin Heidelberg 2006

of these knowledge acquisition/elicitation techniques are visual or graphical. The traditional card sorting and laddering methods are, however, extremely difficult to manage and track back – you will find it nearly impossible to keep the record for hundreds of cards or paper pieces and go back to a prior status without a complicated series of actions, such as video tape recording, searching and playing back.

A key motivation for our work comes from the CO-ODE project[1] where we experienced a rather big gap between manually applied knowledge elicitation techniques, in particular card sorting and laddering, and building of ontologies with Protégé. Protégé [4] is one of the most popular ontology editors which supports many of the tasks of ontology engineering. Protégé enables users to create ontologies by defining concepts, specifications, relationships, annotations and other information within a certain domain. In our CO-ODE project tutorials users wanted to build a pizza ontology with the Protégé OWL Plug-in [5], however, often our tutees preferred to write the pizza terminologies and properties in a pile of cards and to construct the conceptual taxonomy by arranging the cards on the table. After that, the users recorded the outcome in Protégé by manual transfer from the hard copy on their real desktop. It turned out to be very common that users were getting confused and found it difficult to manage a whole table of cards. Finally, any re-sorting of the cards required careful adjustment of the ontology modelled in Protégé.

Our core contribution consists of a novel technique for integrating the KE techniques of card sorting and laddering into ontology building (cf. Section 3). The technique has been implemented in a corresponding plug-in for Protégé (cf. Section 5) that allows graphically eliciting knowledge from documents using card sorting and laddering approaches. Our aim is to seamlessly integrate the KE techniques into the ontology building process to make ontology building more efficient and less error-prone. As a side-effect the persistent storage of card sorting and laddering results allows for later traceability. KE largely depends on user interaction with the plug-in, therefore we employed user-centred design principles to capture requirements (cf. Section 4). After implementation the plug-in was evaluated thoroughly against the requirements (cf. Section 6). The evaluation shows that this KE plug-in meets many of the user's expectations and indeed saves them considerable time when building ontologies.

2 Related Work and Challenges

Related work includes two major aspects. The knowledge elicitation techniques deal with the theoretical issues, while the ontology engineering aspect aims to facilitate users in the process of building ontologies.

From the mid 1980s, people began to do research on expert systems as a subdiscipline of Knowledge Engineering, and it was also the starting point for the scientific research on KE. People tried to develop KE techniques to get knowledge with effectiveness, efficiency and correctness. A number of these methods were borrowed from cognitive science and other disciplines such as Anthropology, Ethnography, and Business Administration [6,7]. KE techniques were effectively used in early 1990s, with the popularity of graphical based personal computer system [8].

[1] `http://www.co-ode.org`

Card sorting is a comprehensive technique of knowledge elicitation methods and is now being used in several disciplines such as Knowledge Engineering, Psychology, and Marketing. In the field of KE, card sorting is considered to be one of the most effective ways for eliciting the domain expert's idea about the knowledge structure. Much evidence shows that card sorting has many positive aspects in making a useful and reasonable elicitation experiments, including helping the respondents to recall the domain concepts; providing a structuralized concepts pile for future processing – such as laddering; fast acting and easy handling [1,8]. Figure 1 shows a real use case of card sorting.

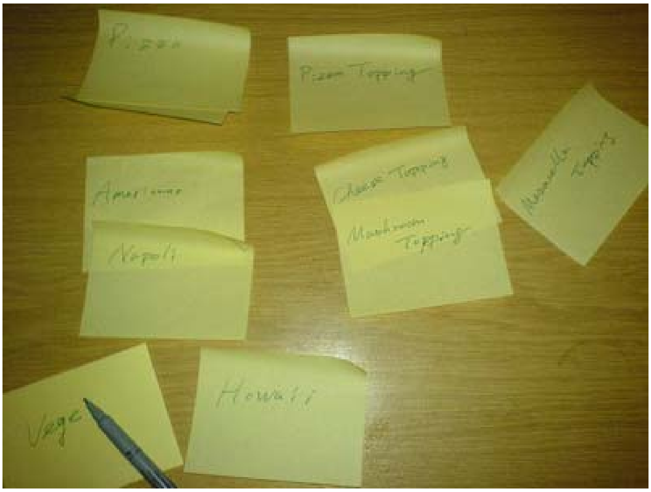


Fig. 1. Traditional card sorting

Laddering has been widely used in the field of knowledge elicitation activities in recent years. The basic purpose of the laddering method is to elicit people's goals and values [8,9]. People from knowledge elicitation community have developed a well-established range of formal semantics, procedures and notation for building ladders. Based on the Rugg and McGeorge's [9] categorisation, laddering can be used for three major purposes – they are using laddering to elicit sub-classes, explanation, goals and values. Laddering has been implemented as an independent tool in a software toolkit, called PCPACK [10] with integration of the CommonKADS method [11]. But the implementation in PCPACK lacks the compatibility with full support of the OWL language, and the well integration with state-of-the-art ontology engineering tools, like Protégé. In this paper, we are going to use the laddering method to build up the data and object properties for concepts, e.g. a typical conceptual ladder example is shown in Figure 2.

Not many general purpose approaches have been invented for building ontologies. The available methodologies are generally frameworks with descriptions and outlines on how to build ontologies [12,13]. Often, such methodologies focus on the ontology engineering steps without much support for the very early stages of KE. The most re-

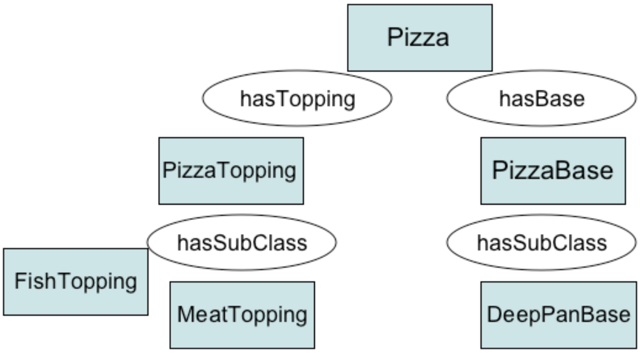


Fig. 2. Concepts map with properties

cent methodology DILIGENT [14] supports the dynamic nature of ontologies and also includes support for argumentation between different actors during the whole process.

The challenges tackled in this paper, therefore, are how to implement a knowledge elicitation methodology which extends existing ontology engineering methodologies and at the same time focuses on the development of a usable tool that supports graphically-oriented building of ontologies, using card sorting and laddering tools respectively, and which allows people to further refine their ontology in a (broader) ontology editor like Protégé.

Our technique and system aims to reduce the work-load for knowledge engineers and domain experts; increase the reusability of laddering and card sorting processes; effectively manage the KE tasks; and seamlessly integrate with an existing software system for ontology engineering.

3 KE Integration Technique

The traditional card sorting method generally consists of a pile of cards with the approximate size of a credit card, created by the researchers, who write or print the domain concepts on cards. A video tape recorder captures both the acts and voices of the entire procedure for future analysis. We can therefore find out that traditional card sorting has three major drawbacks, which are, easy to be destroyed, difficulty to be managed, and not practical to be transferred to computer files. For example, a blast of wind or a cup of coffee can easily disrupt the process, this manual process also cannot be shared on the internet, and the video information is also a bottle-neck while people have many tapes to manage or deliver via the internet.

The solution is straightforward. To avoid the fragility of actions, we can transplant the entire procedure into the computer, and by using a preliminary version-control mechanism, the user can overall control their milestones when they sort cards and structure concepts. Obviously, this plug-in does not record the activities by capturing the screen just like a screen recording software, contrarily, it logs the activities performed by the user in a text file which is essentially easy to be transferred and managed.

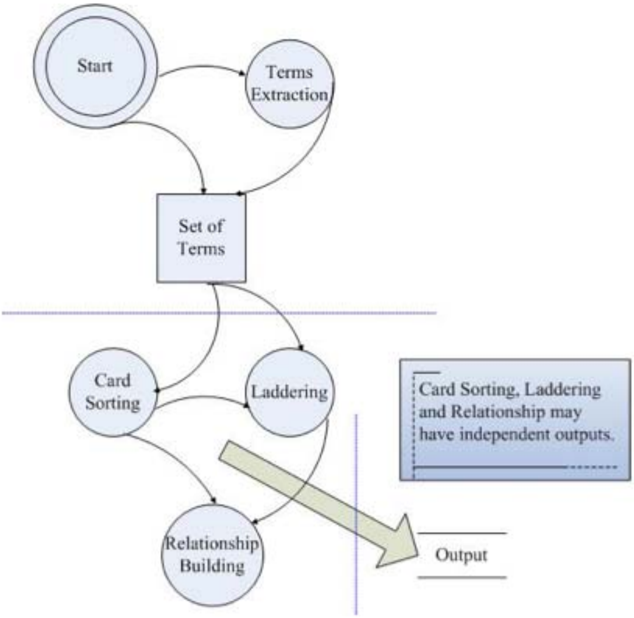


Fig. 3. KE integration technique roadmap

Redo and undo mechanisms in text editors give us a hint to solve this problem by automating the tasks. The whole procedure and all its related matters – we call it version control manager – need to be temporarily stored in the memory and saved to the permanent storage devices if necessary. By doing this, the domain experts and developers are able to go back to anywhere if they want, all they need to do is to store the different versions of the tasks while they are standing at a milestone or a trap point.

Laddering techniques play an important role in discovering the potential relationships between the domain concepts. The laddering method is usually used combining with other KE methods such as card sorting. The subjects and objects within the ontology are inter-connected with several kinds of relationships elicited from the domain experts via laddering, and the structural source of subjects and objects are built via card sorting. As ontology is the structuralized domain knowledge base generated from experts, we can realise that laddering is undoubtedly essential while developing ontologies. So we are also going to implement the laddering technique as part of this plug-in, as well.

Figure 3 illustrates the process of applying KE techniques as part of ontology building. As indicated in the figure there are multiple ways to apply the various KE techniques. In the following we briefly explain three frequently used ways to apply the process.

1. Start with a [Set of Terms] and perform card sorting and/or laddering
2. Start with a term extraction to retreive a [Set of Terms], then perform card sorting and/or laddering
3. Start with a term extraction to retrieve a [Set of Terms], perform card sorting and/or laddering, then perform relationship building

4 Design and Implementation

While we want to build **real usable** software, rather than a program for demonstration purposes, there are many principles to be followed, especially those related to the design of the user interface. The implementation phase is tightly coordinated with the interface design and there are many rolling procedures to refine the design and implementation respectively.

4.1 User-Centred Design

In the Human-Computer Interaction (HCI) research field, user-centred design, also known as usability engineering, is one of the most central methodologies and now widely used in various disciplines, including Software Engineering, Knowledge Management or Information System [15].

One of the objectives of the CO-ODE project is to provide a user-oriented tool set for the Protégé OWL Plug-in, so the user-centred design techniques will be kept in mind throughout the entire plug-in design life cycle.

A key aspect of user-centred design techniques is to make users involved in the software design process, by interviewing various groups of users based on certain requirements, such as age, occupation, gender, culture and so on. The interview results will be gathered and analysed in order to discover the goals and values of the target user group. The techniques of user participatory design are obligatory while designing this KE plug-in, because the target user group is mainly scientific researchers with different disciplines, requirements, personalities, ways of working and thinking.

User participant design includes observing and recording the manual activities, such as using the paper as window frames; cutting the paper into rectangles with difference size as dialogues and menus; choosing difference colours as different selection feedback; drawing, dragging while necessary to modify the interface; taking the picture while performing activities and many other actions. All these are performed by the **real** target group of users. The picture below was taken from the interview activities within the design process of this plug-in.

It is thereby necessary to set a predefined series of interviews in which we invite potential users to participate, and so that we can collect design information. Some interview methods such as unstructured interview and structured interview are going to be employed for different purposes.

An unstructured interview usually tends to be used in the early stages of the interview session, in which the users will be asked some general questions. In this plug-in design, at first, we need to know the user's general points of view about the card sorting and laddering tools, the user's attitudes towards the perspectives of this plug-in and perhaps, and their personal manners of using computer software. Unstructured interview results will provide the developers with appropriate concepts and sensible ways of thinking, rather than the technical details.

Comparatively, it is much easier to hold an interview with a list of predefined questions. The structured interview design is more important for the software designers because all the interviewees will be asked a same set of questions related to the software technical details. The analysis of the structured interview results are crucial since the

Fig. 4. A user participant design case

detailed technical issues in the software design phase will be addressed based mainly on these results.

4.2 Case Study of Interviews

The interview results show remarkable differences between people with different academic backgrounds, however, it also shows that the age, gender, and cultural background don't play essential roles. Probably that's because of the statistical analysis requires a much bigger sample, but we have already collected enough information required for the design of this KE plug-in.

Learning from the interview results, this software system should have 1) a input from document and elicitation functionality on user interface; 2) a series of cards generated from the text with round rectangle and the colour style of Protégé, that's because using the colour style of an existing popular base system tends to be more acceptable; 3) a flexible and straightforward user interface with layout of placing the working panel – both the card sorting and laddering tool, at the left as tabbed widgets, and putting the operation results on the right, as well as a number of buttons reasonably arranged; 4) a well-formatted output.

4.3 Implementation

Now we can conclude the procedure of building this plug-in with user-centred design techniques involved, which can be displayed in Figure 5 in a straightforward manner.

From Figure 5 we can find that the interview sessions should be held while interviewees are performing and modeling the KE techniques – basically, the card sorting and laddering techniques, and the design of the user interface, i.e. the 3D rectangle objects are issues related to the HCI area. User participant design methods are also applied in the testing and evaluation sessions. We can see from the figure that there might be some

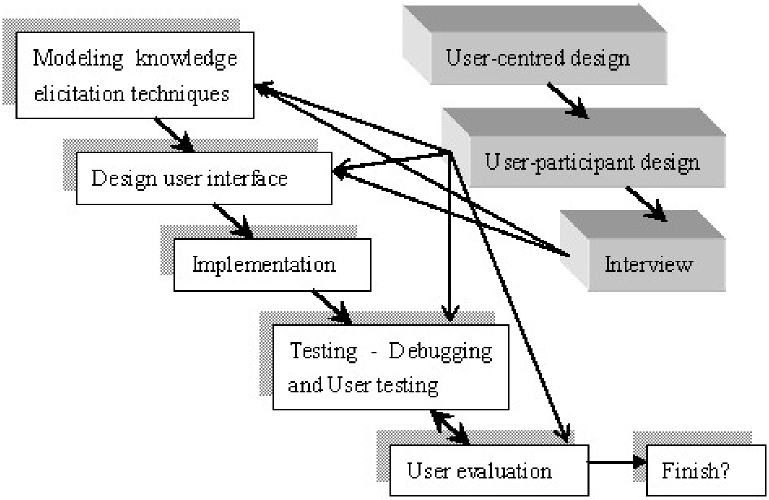


Fig. 5. Whole picture of design and implementation

loops between the testing/debugging and user evaluation circle, which is because of the fundamental software engineering rule – developers never know when the project will be finished and what kind of extensions should be added. It depends on the evaluation results and up-to-date user requirements.

5 Application of the Plug-In

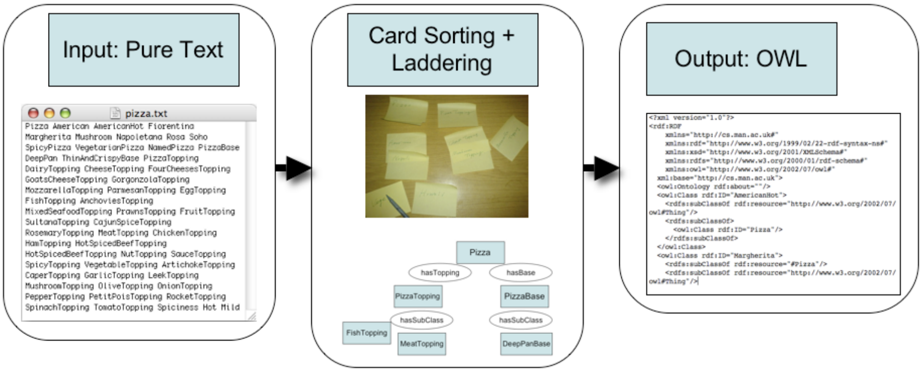
We first built a prototype with a focus on application input/output aspects, then the prototype was extended to the existing Protégé system. Protégé quite naturally was our first choice as underlying infrastructure due to its widespread adoption and its easy extensibility. There already exists a plethora of freely available plug-ins that extend the basic functionalities.

5.1 Prototype

The KE methodology (cf. Figure 3) requires input and output, in which there are some trade-offs between the simplicity of the user interface and strength of functionality.

For the input, to show the most straightforward idea of this plug-in, we are going to use plain text as the source of concepts. There are many completed projects investigating how to use text mining and natural language processing (NLP) techniques to acquire knowledge from text, for instance, Text2Onto project [16]. So obviously, to search for a possible extension with existing tools is a better choice rather than developing a new one.

The format of the output is one of the most important design issues, because a primary consideration of this plug-in system is extendibility, which emphasises globally unified input/output. This software system might have many possibilities of input, thus we are going to discuss the output format here.



```
Pizza American AmericanHot Fiorentina
Margherita Mushroom Napoletana Rosa Soho
SpicyPizza VegetarianPizza NamedPizza PizzaBase
DeepPan ThinAndCrispyBase PizzaTopping
DairyTopping CheeseTopping FourCheesesTopping
GoatsCheeseTopping GorgonzolaTopping
MozzarellaTopping ParmesanTopping EggTopping
FishTopping AnchoviesTopping
MixedSeafoodTopping PrawnsTopping FruitTopping
SultanaTopping CajunSpiceTopping
RosemaryTopping MeatTopping ChickenTopping
HamTopping HotSpicedBeefTopping
HotSpicedBeefTopping NutTopping SauceTopping
SpicyTopping VegetableTopping ArtichokeTopping
CaperTopping GarlicTopping LeekTopping
MushroomTopping OliveTopping OnionTopping
PepperTopping PetitPoisTopping RocketTopping
SpinachTopping TomatoTopping Spiciness Hot Mild
```

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns="http://cs.man.ac.uk#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://cs.man.ac.uk">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="AmericanHot">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/
owl#Thing"/>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Pizza"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Margherita">
    <rdfs:subClassOf rdf:resource="#Pizza"/>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/
owl#Thing"/>
```

Fig. 6. The plug-in prototype

Basically, the proposed output file formats are: pure text, HTML and XML/RDF. Pure text is the most common way to store information, however, it may have different default format like ASCII or Unicode, while they are processed in different operating systems. HTML is a well-defined syntax-based mark-up language and easy to be parsed, but an HTML file is not easly machine-understandable. The Resource Description Framework (RDF) [17] is based on XML technology with machine readable format, and the processing of RDF is well-implemented by many third party programming language APIs.

As matters stand above, the most sensible choice is to use RDF for information storage in this software because of the consideration of feasibility, portability and acceptability. Another possible output is to use the existing Protégé components such as Protégé OWL Plug-in to directly transfer the output to the ontology tree for future development.

After making the decision about the input/output issues, we have the prototype for this plug-in as in Figure 6.

The starting point is we have a series of terms in a text format file. The users can initialise their ontologies by card sorting and laddering as tool tabs in the software interface at the middle, and then at last the users get the output as an RDF document. Thereafter by applying the prototype demonstrated, the plug-in can be developed in reality.

5.2 The Reality

Assume that users have a source of texts, in which there are a list of terms, this plug-in allows a user to elicit terms from pure text as Figure 7 shows. After the elicitation procedure, the users can build a subsumption relationship conceptual tree from the working panel, by adding, deleting and editing the cards, which is controlled overall by the "Version Manage", marked in Figure 7.

Essentially, this is a typical **card sorting** session. The two processes mentioned are combined together as the conceptual modeling for generating the subsumption relationship, depicted in Figure 7, respectively. The black colored numbers is to indicate the steps of operations. The blue arrows will show as dark gray if the paper is not colour-printed.

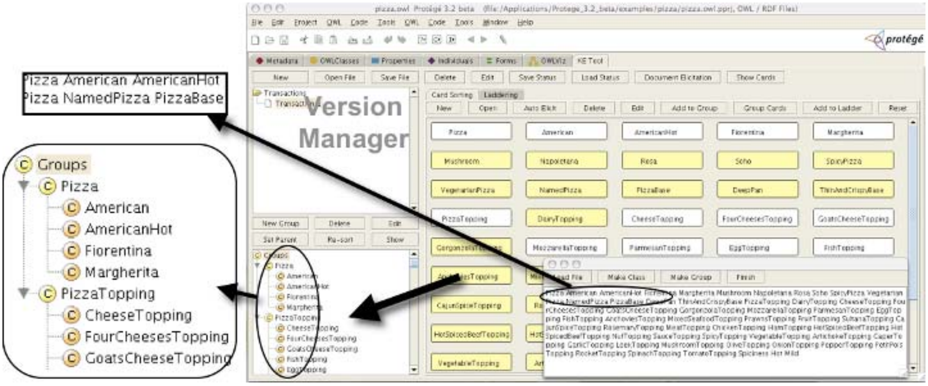

Pizza American AmericanHot
Pizza NamedPizza PizzaBase
Groups
Pizza
American
AmericanHot
Fiorentina
Margherita
PizzaTopping
CheeseTopping
FourCheesesTopping
GoatsCheeseTopping
pizza.owl Protégé 3.2 beta (file:/Applications/Protege_3.2_beta/examples/pizza/pizza.owl.pprj, OWL / RDF Files)
File Edit Project OWL Code Tools Window Help
protégé
Metadata
OWLClasses
Properties
Individuals
Forms
OWLViz
KE Tool
New
Open File
Save File
Delete
Edit
Save Status
Load Status
Document Elicitation
Show Cards
Transactions
Version Manager
Card Sorting
Laddering
New
Open
Auto Elicit
Delete
Edit
Add to Group
Group Cards
Add to Ladder
Reset
Pizza
American
AmericanHot
Fiorentina
Margherita
Mushroom
Napoletana
Rosa
Soho
SpicyPizza
VegetarianPizza
NamedPizza
PizzaBase
DeepPan
ThinAndCrispyBase
PizzaTopping
DairyTopping
CheeseTopping
FourCheesesTopping
GoatsCheeseTopping
GorgonzolaTopping
MozzarellaTopping
ParmesanTopping
EggTopping
FishTopping
AnchoviesTopping
CajunSpiceTopping
HotSpicedBeefTopping
VegetableTopping
New Group
Delete
Edit
Set Parent
Re-sort
Show
Load File
Make Class
Make Group
Finish
Pizza American AmericanHot Fiorentina Margherita Mushroom Napoletana Rosa Soho SpicyPizza Vegetarian Pizza NamedPizza PizzaBase DeepPan ThinAndCrispyBase PizzaTopping DairyTopping CheeseTopping FourCheesesTopping GoatsCheeseTopping GorgonzolaTopping MozzarellaTopping ParmesanTopping EggTopping FishTopping AnchoviesTopping MixedSeafoodTopping PrawnsTopping FruitTopping SultanaTopping CajunSpiceTopping RosemaryTopping MeatTopping ChickenTopping HamTopping HotSpicedBeefTopping HotSpicedBeefTopping NutTopping SauceTopping SpicyTopping VegetableTopping ArtichokeTopping CaperTopping GarlicTopping LeekTopping MushroomTopping OliveTopping OnionTopping PepperTopping PetitPoisTopping RocketTopping SpinachTopping TomatoTopping Spiciness Hot Mild

Fig. 7. Building subsumption tree

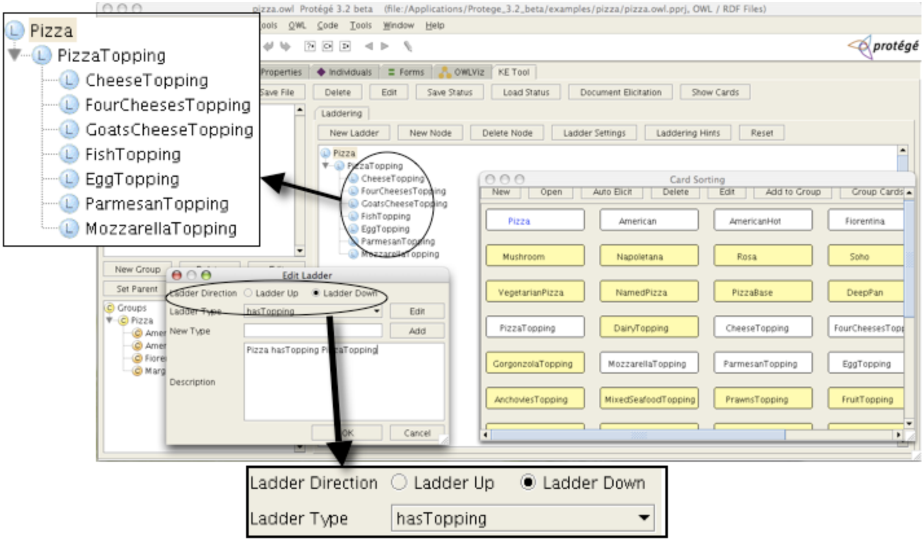
The prototype figure 6 has displayed the steps to elicit knowledge from text. A pure text window which locates at the right bottom of the figure is providing the knowledge resource. The users can load the text to the working panel that consists the cards to be sorted. Then the users can begin to sort the cards into different piles or groups that are displayed as the tree showed at the left of Figure 7 to get the subsumption tree as the very first ontology structure. Different levels of colors indicate the status of each card, illustrating whether the cards have been sorted (white) or not.

Card sorting panel gives users a clear and straightforward illustration of how the concepts are arranged and which concept has or has not been sorted. By using this component, the users are not going to be confused by the texts listed on the documents but sorting fast and correctly.

While the users have the skeleton of the ontology and they want to add properties between the concepts, the **laddering** tool provides a smooth route for this task. The detailed procedures are indicated at Figure 8. The card sorting window is not flowing at the right bottom, showing the procedure of how to add concepts from a series of cards to the ladder. The conceptual ladder with magnified view locates at the centre, and we can observe from the magnified ladder edition windows at the bottom that the relation between the current concepts "Pizza" and "PizzaTopping" is "hasTopping". The direction of the current ladder is "ladder down", therefore the domain of object property "hasTopping" is "Pizza", while class "PizzaTopping" is the range. In our integrated KE technique, we build object properties in this way.

To control the card sorting and laddering tools overall, and to enable users to easily manage their milestones, a version manager function is implemented by saving and loading the runtime status of the plug-in, both the information of the laddering and subsumption structure, as well as the contents in card sorting tab, to and from the main memory.

We make use of this manager to organise the global actions performed by the users so that users are able to track back to their previous task runtime status by simply choosing and loading the different versions that are created and saved before. The users just need to choose a target version, press a "Save Status" button, and then the system will give a message to tell the users whether the version is successfully saved or not. Once the users



Ladder Direction ○ Ladder Up ● Ladder Down

Ladder Type hasTopping

Pizza hasTopping PizzaTopping

Fig. 8. Laddering

come back from other versions, they can simply load this version into the working tabs and trees immediately by pressing the "Load Status" button. This component lays on the upper right side of the interface, which can be seen from the screen shots in Figure 7 with label "Version Manager".

6 User Evaluation

User evaluation shows the user's attitudes towards the quality of this software. Based on the requirements of user-centred design, the feedbacks from user evaluation will be treated as an essential guideline for software testing and debugging.

We evaluated our approach by performing a user evaluation based on the requirements of the user-centred design. The user evaluation has two different parts. One is the interface evaluation which concerns the GUI, including ease of use, look and feel, and so on. And the other one is functional evaluation whose emphasises are the background functionalities. This evaluation methodology aims to detect the user's comments on two basic aspects in the domain of user-centred design – the software should be powerful, flexible and robust.

6.1 Evaluation Result

There were eleven people involved in the user evaluation activities, and they are diverse in academic and cultural backgrounds. In order to quantify the result, a grading system similar to the university examination was borrowed (0-10 scoring scale), that is, 5 is a pass, 6 is a good pass, and above 7 is a distinction. In the arrays of the scores introduced below, the first five scores in each array come from the experts or frequent users of knowledge systems. The participants are marked with “E” for expert and “N” for “Non-expert” plus the reference number.

In terms of the user interface design, the grading result will be given to four different aspects as **interface evaluation**. The users were asked for the grades of the four points, and their grading results are listed in Table 1. To be statistically accurate, the average score was calculated by eliminating the highest and the lowest scores in each array.

Table 1. Interface evaluation results

Participant	E1	E2	E3	E4	E5	E6	N1	N2	N3	N4	N5	Average
Look and feel	9	7	7	8	9	7	7	6	7	8	6	7.3
Interface layout	7	7	9	6	8	6	9	5	6	5	6	7.3
Ease of use	7	7	6	7	8	7	6	6	6	6	5	6.4
Flexibility	7	8	8	6	5	6	6	6	9	6	8	6.8

The overall score was calculated by formula using standard deviation so we get **6.7** points here.

The **functional evaluation** involved the grading of each basic component, including card sorting, laddering, relationship setting and version manager. They are four major components provided by this plug-in and users are easily getting familiar with them, so the grading of these components is direct.

Table 2. Functional evaluation results

Participant	E1	E2	E3	E4	E5	E6	N1	N2	N3	N4	N5	Average
Card sorting	9	8	7	9	9	7	8	8	7	7	8	7.9
Laddering	7	6	7	7	7	7	6	7	7	9	8	7.0
Relationship setting	6	7	6	6	6	8	7	9	7	8	8	7.0
Version manager	9	9	8	8	9	8	7	8	9	8	9	8.4

We can see that the overall is **7.6** points. After taking the scores, we now analyse the results and make a conclusion.

6.2 Result Analysis

From the scores, we can simply find out that the users are mostly satisfied with the functionalities of the plug-in, which stands that the primary user-centred design procedure has been well-established. With respect to the interface of this plug-in, although the score is comparatively moderate, the users also generally have given positive comments.

To discover more from the evaluation results, we find that the interface look and feel, card sorting and version management components have the highest ratings and are thought to be the best implemented. Meanwhile, the elements related to the ease of use principle and interface layout arrangement require much future improvement.

If we go further, we may find that the plug-in interface are more appreciated by the experts rather than the amateurs, because the knowledge engineering experts are more familiar with the existing Protégé system, card sorting and laddering approaches. They find that this software have a unified style with the Protégé system, which doesn't quite make sense to the non-experts, though. Otherwise, contrarily the experts are not fully satisfied with the laddering tool and relationship setting component. Their feedback express the way of their working is somewhat different from how this plug-in does. That's because, people from different disciplines are likely to use laddering tool in many different ways for different purposes, and the plug-in is developed according to the design principles of CO-ODE project with strong emphasis in the medical and biological domain.

It is worthwhile to mention that the evaluation of the software from the wholly independent UK Freshwater Life Biological Association, and their comment on this software is:

> *"It was good to see what he has been doing and looks like a potentially very useful tool. We really liked to get our hands on a copy to play around with. Even in its current state it could save us considerable time."*

In a nutshell, this plug-in is commonly considered to be a well-implemented and powerful tool in **real** use, whereas the interface is possibly only recognised by the knowledge system experts. All the evidences in this user evaluation procedure show that people are very eager to see the future development of this plug-in.

7 Conclusion and Outlook

We presented a technique for supporting the building of ontologies by using and integrating the knowledge elicitation techniques of card sorting and laddering. We developed a Protégé plug-in by employing user-centered design methods and thoroughly evaluated the research outcome. In the evaluation users performed very well with the plug-in and gave highly valuable feedback for future development. The conventional KE techniques were seamlessly integrated to the ontology building process to close the gap in the traditional manual ontology engineering cycle.

Future work includes the extension of the plug-in with some featured capabilities from the Text2Onto tool [16] to automatically extract terms from large texts using text mining and ontology learning techniques to further heuristically speed up the ontology building process.

Acknowledgements

The research reported in this paper was supported in part by the CO-ODE project (`http://www.co-ode.org/`) funded by the UK Joint Information Services Committee and the HyOntUse Project (GR/S44686) funded by the UK Engineering and Physical Science Research Council and by 21XS067A from the National Cancer Institute. The authors' current research is also supported by the EU-IST-506826 SEKT project (`http://www.sekt-project.com`). This publication only reflects the authors' views. We would like to thank the support from the potential users and our colleagues for fruitful discussions.

References

1. Upchurch, L., Rugg, G., Kitchenham, B.: Using card sorts to elicit web page quality attributes. IEEE Software **18** (2001) 84–89
2. Cooke, N.J.: Varieties of knowledge elicitation techniques. Int. J. Hum.-Comput. Stud. **41** (1994) 801–849
3. Hinkle, D.: The change of personal constructs from the viewpoint of a theory of construct implications. PhD thesis, Ohio State University (1965) Cited in: Bannister, D. and Fransella, F. (1980). Inquiring Man. Penguin, Harmondsworth.
4. Noy, N.F., Sintek, M., Decker, S., Crubézy, M., Fergerson, R.W., Musen, M.A.: Creating semantic web contents with protégé-2000. IEEE Intelligent Systems **16** (2001) 60–71
5. Knublauch, H., Fergerson, R.W., Noy, N.F., Musen, M.A.: The protégé owl plugin: An open development environment for semantic web applications. In: International Semantic Web Conference. (2004) 229–243
6. Boose, J.H.: Knowledge acquisition techniques and tools: Current research strategies and approaches. In: Proceedings of Fifth Generation Computer Systems. (1988) 1221–1235
7. Hoffman, R.R.: The problem of extracting the knowledge of experts from the perspective of experimental psychology. AI Magazine **8** (1987) 53–67
8. Shadbolt, N., Hara, K.O., Crow, L.: The experimental evaluation of knowledge acquisition techniques and methods: history, problems and new directions. International Journal of Human-Computer Studies **51** (1999) 729–755
9. Rugg, G., Eva, M., Mahmood, A., Rehman, N., Andrews, S., Davies, S.: Eliciting information about organizational culture via laddering. Journal of Information System **12** (2002) 215–230
10. Milton, N.: PCPACK Toolkit. (2003) www.epistemics.co.uk/Notes/55-0-0.htm.
11. Schreiber, G., Wielinga, B.J., Akkermans, H., de Velde, W.V., Anjewierden, A.: CML: The CommonKADS conceptual modelling language. In: Proceedings of 8th European Knowledge Acquisition Workshop (EKAW). (1994) 1–25
12. López, M.F., Gómez-Pérez, A., Sierra, J.P., Sierra, A.P.: Building a chemical ontology using methontology and the ontology design environment. IEEE Intelligent Systems **14** (1999) 37–46
13. Sure, Y., Staab, S., Studer, R.: On-to-knowledge methodology. In Staab, S., (eds.), R.S., eds.: Handbook on Ontologies. Series on Handbooks in Information Systems. Springer (2003) 117–132
14. Tempich, C., Pinto, H.S., Sure, Y., Staab, S.: An argumentation ontology for distributed, loosely-controlled and evolving engineering processes of ontologies (diligent). In Gmez-Prez, A., Euzenat, J., eds.: 2nd European Semantic Web Conference (ESWC 2005. Volume 3532 of LNCS., Heraklion, Crete, Greece, Springer (2005) 241–256
15. Shneiderman, B.: Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1997)
16. Cimiano, P., Völker, J.: Text2onto – a framework for ontology learning and data-driven change discovery. In: Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB'05). (2005)
17. Lassila, O., Swick, R.: Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation,World Wide Web Consortium, Boston. (1999) www.w3.org/TR/REC-rdf-syntax (current 6 Dec. 2000).

Towards a Topical Ontology of Fraud

Gang Zhao[1] and Robert Meersman[2]

[1] Intelartes, BP88, Bruxelles 4, Belgium
Gang.Zhao@intelartes.com
[2] STARLab, Computer Science, Vrije Universiteit Brussel, Belgium
Robert.Meersman@vub.ac.be

Abstract. The paper describes the concept of *topical ontology* and the development of a topical ontology of fraud. A topical ontology is concerned with a set of themes identified to represent the knowledge structure of the domain expert. It reflects the specific scope, perspectives and granularity of conceptualization about the themes. It sits on the base ontology, integrates multiple domain ontology and serves as the knowledge framework for application ontologies. It is architected with a basic conceptual schema and configuration design pattern to capture the knowledge structure as well as concepts and relations of the knowledge.

1 Introduction

FF POIROT (www.ffpoirot.org), an EC funded IST project, explores the use of the ontology technology for fraud prevention and detection. Its ontology engineering is one of the key motivations towards the idea of *topical ontologies* and the associated methodology of its design and development. The ontology of financial fraud was initially built from actual fraud cases and violated legislation of stock exchange regulation. A case-driven and bottom-up approach, using the ontology engineering methodology AKEM [9], resulted in an application ontology of 800 relationship types, not counting their specific constrained or instantiated instances.

The need for further development and reuse revealed an inadequacy in the ontological model: it is an haphazard representation of the essentials in the knowledge structure about financial frauds. To resolve it, the upper-level and domain ontologies were resorted to, in order to see if they can help structure the model. 186 relevant concepts are selected from the SUMO [7] base and domain ontologies, 123 object-role relationships are introduced for alignment. The resultant model did not exhibit a systematic conceptual framework of knowledge about frauds. Three major problems are a) missing essential concepts, b) mismatched granularity and abstraction between related objects and relationships; c) implicit assumption of particular perspective or conceptual scope of objects and relationships. Consequently the base/domain ontologies and application ontologies 'wired' ad hoc. The conclusion is that neither the base nor domain nor application ontologies represent well the perspectives, scope, abstraction and granularity in the conceptualization in the knowledge structure of experts on the subject. Architecturally, there is a missing layer between application ontologies, on the one hand, and base and domain ontologies, on the other. It should serve to

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 566 – 572, 2006.
© Springer-Verlag Berlin Heidelberg 2006

capture the common viewpoints, principles and patterns in this particular expertise shown in [1] [8].

2 Topical Ontology

A topical ontology is an ontology about a set of themes. It is intended to represent the knowledge structure of domain experts: the scope, assumptions, principles, rules and patterns concerning the themes, for example, intellectual property rights [2]. It consists of abstract objects and relations. It also represents how these objects and relations are clustered to reflect the knowledge structure. In FF POIROT, topical ontologies are built for descriptive and operative rather than prescriptive purposes [10].

It is not an upper or base ontologies [5] [7]. Its contents are not necessarily universal concepts with generic axioms. Instead it reflects a particular cut of knowledge space for specific relevance, in a particular angle of observation and granularity. It is not a domain ontology, as financial or economic ontologies [7]. Nor is it the subset or extension of a domain ontology, since the themes represented can cut across multidisciplinary subjects. The concept of fraud is a good example. It is not an application ontology. Its purpose is descriptive, not to capture operational details relevant to a particular task performed by a given fraud examiner in a given system context. The topical ontology bases itself on the upper or foundational ontologies, restructures multiple domain ontologies and lays foundation for application specific ontologies.

The scope of the topical ontology is determined with respect to a set of user stakeholders. For example, the actors in the lifecycle of frauds such as intelligence analyst, fraud investigator and prosecutor, are the stakeholders of a topical ontology of financial frauds. They fix the scope and viewpoint of the conceptual domain. In other words, a topical ontology represents given perspectives of knowledge, specific to the stakeholders' conceptualization of the themes.

3 Architecting Topical Ontology

The chosen viewpoint ensures the management of semantic scope and perspectives. Next is how to represent explicitly the ideational structure in the scope from the given perspective. Two structural principles are explored: *framework of analysis* and *design patterns*.

3.1 Framework of Analysis

The framework of topical ontology analysis partitions the model space in the domain and application perspectives. The domain perspective captures *fact types* [3]. A fact type consists of the object(s) and their role(s), and their ideational context. It is called lexons in the *DOGMA* ontology framework [4] [6] (see Table 1 for example).

The application perspective captures specific constraints, instantiations and verbalization of lexons and their connections in a specific application and system context. The distinction of the two layers of the ontology model modularizes the universe of semantics for change encapsulation, model scalability and reusability [10].

Table 1. Examples of lexons

Context	Term	Role	Term	Role
FraudElement.FD.01	Perpetrator	Gain	Assets	GainedBy
FraudElement.FD.01	Perpetrator	Gain	FinancialInstrument	GainedBy
FraudElement.FD.01	Perpetrator	Gain	Material	GainedBy
FraudElement.FD.01	Perpetrator	Misrepresent	Information	MisrepresentedBy
FraudElement.FD.01	Perpetrator	Falsify	Information	FalsifiedBy
FraudElement.FD.01	Perpetrator	Withhold	Information	WithheldBy
FraudElement.FD.01	Perpetrator	Deceive	Victim	DeceivedBy
FraudElement.FD.01	Perpetrator	Convert	Assets	ConvertedBy
FraudElement.FD.01	Perpetrator	Convert	Material	ConvertedBy

3.2 Design Pattern

The topical ontology is architected from a set of topical concepts. Design patterns are used to configure the ontological representation of a topical concept and its relationship with other concepts.

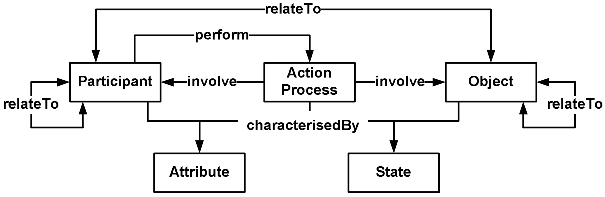


Fig. 1. Basic conceptual schema

3.2.1 Basic Conceptual Schema

A topical concept is analyzed in terms of a basic conceptual schema of six entities (*Participant, Action or Process, Object, Attribute, State*) and four relations (*perform, involve, relate to, characterised by*), as indicated in Fig. 1. The core concepts of a topical ontology are typically concerned with entities: *Participant, Action/Process* and *Object*. For example, a *Participant*-centred topical ontology of business processes presents an organizational view. An *Object*-centred describes a data or product view. An *Action/Process*-centred captures a process-event view.

3.2.2 Topic Configuration Pattern

The design pattern for topical ontologies is of two purposes: explicit organization by the architecture style or pattern and the facilitation of consensus building to integrate different views and conceptualization. The development of the topical ontology of fraud uses the topical concept pattern in Fig. 2.

The *Schema* and *Element* represent the basic conceptual schema discussed in Fig. 1. The *Schema* and *Configuration* are the conceptualization pattern to describe a topical concept. The *Configuration* and *Component* constitutes a specific (partial or full)

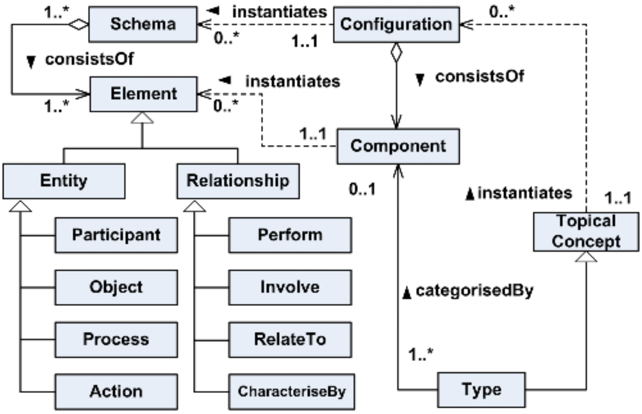


Fig. 2. Ontology design pattern: topic configuration

instantiation of the basic schema, capturing the perspective of conceptual analysis. The *Topical Concept*, with its conceptual blueprint determined by the *Configuration*, represents the focal concept to model in the topical ontology. For example, the model of actors in business processes can be conceptualized on the schema entity, *Participant*. The *Configuration* of this topical concept can be designed with a choice of *Elements* to be the *Components*. The topical concept can be classified in terms of particular aspects represented by the *Component* of the *Configuration*.

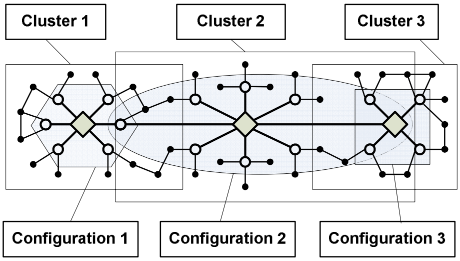


Fig. 3. Topological illustration of topical concept, topical cluster, configuration

3.2.3 Architecture of a Topical Ontology

The design pattern is applied to organize ontological objects into modules: *topical clusters*. A topical cluster is a set of object-role relationships or lexons, grouped conceptually around a topical concept. The *configuration* represents a meta conceptual structure of which objects and roles are included and how they are related with each other. The *configuration* of the topical concept is also the mechanism to capture the meta-level relationship between topical clusters in the form of inter-configuration relationship.

Fig. 3 shows an illustrative topological structure of three topical clusters indicated in boxes, and their corresponding configurations in the hexagon, oval and square shades. The vertices are conceptual objects and edges are roles. There are three types of vertices, topical, configurational and peripheral by their function with respect to the topical concept. The diamond vertices are topical point of the cluster. The circle vertices are configurational concepts, which constitute the configuration of the topical concept. The solid vertices are peripheral, the inessential but relevant concepts. The figure shows three cases of inter-configuration relationship. Firstly, *Configuration 1* and *2* anchor on the same conceptual object as their respective configurational vertices. Secondly, the configurational vertex in *Configuration 2* is topical in *Configuration 3*. Thirdly, two configurations are connected through the peripheral vertices, shown between *Configuration 1* and *2*, between *Configuration 2* and *3*.

3.2.4 Representing the Knowledge Structure of Domain by Topical Configuration

The topical ontology not only consists of conceptual objects, roles and axioms of well-formedness, but also meta models of how they are grouped and structured into a semiotic system of knowledge. The architecture by topical concepts and their topical configuration enables a systematic approach to both intra-cluster structure of concepts and inter-cluster relation. Such models are not necessarily more abstract or generic metaphysical models as in upper ontologies or collections of common concepts of particular subject domains as in domain ontologies. It embodies the perspective of observation and thematic focus, adopted at a given stage or time of ontology engineering in the understanding of the knowledge structure. It depicts how and where particular concepts are focused on, elaborated or interpreted, whereas the others are ignored or left in coarse-grain sketches. There are no preconceived logical requirements on the boundary, granularity, abstraction level of topical clusters, but their explicit architectural configurations, specific to the conceptual domain.

The selection of topical concepts and design of their configurations is to architect the topical ontology according to the knowledge structure of the domain. Orthogonal to the objects, roles, axioms and constraints, the configurations of the topical concepts serve to capture the knowledge structure for four main purposes in the process of ontology development. Firstly, the ontology can be scaled up with flexibility of multiple perspectives of observation, levels of abstraction and degrees of details by a structured modeling approach. Secondly, the consensus building can be facilitated with explicit and structured description and with the documentation of the knowledge structure. Thirdly, the topical configuration can serve as a map for navigating through the objects and roles within the topical cluster as well as across the clusters. Fourthly, the topical concepts and the explicit specification of their configuration provide both intuitive and structural information for the partial reuse of the topical ontology.

4 A Topical Ontology of Fraud

The topical ontology of fraud is developed in FF POIROT by a multidisciplinary distributed team of investigators, domain experts, knowledge analysts, ontology engineers and application developers. It seeks to capture key concepts underlying hypothesis, reasoning, and interpretation about frauds.

4.1 Overall Architecture

The concept of fraud is the focus of attention of the fraud analyst, investigator and prosecutor and essential semantics underlying the tasks performed manually or automatically. As the central topic of the ontology, it is seen as action rather than object to describe in an associative perspective, instead of classificational perspectives, since the means, location, time, context, participant, objects, motivation, states of the action are essential conceptual parameters for fraud examination and prosecution.

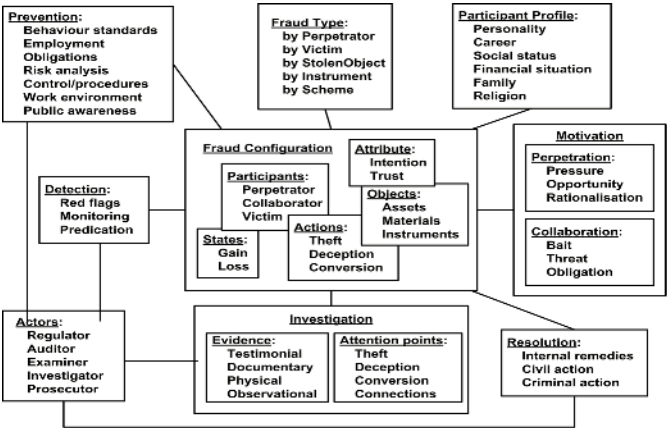


Fig. 4. The architecture of the topical ontology of fraud

Fig. 4. indicates graphically nine key topical clusters. Each box can be seen as subsystem of the ontology. The configuration of the concept of *fraud* is structured in terms of the basic conceptual schema. It is typified by the components of the configuration. The actors and four major processes are identified in the lifecycle of frauds to capture the expertise in fraud prevention, detection, investigation and resolution.

4.2 Topical Clusters

Each subsystem of the ontology (Fig. 4) can be further broken down into modules, using the same configuration design pattern for organization. For example, the fraud types can be further refined as below

I. By Perpetrator
   A. Employee / occupational fraud
   B. Management fraud
   C. Investment fraud
   D. Vendor fraud
   E. Customer fraud
II. By Victim
III. By Stolen Object
IV. By Instrument
V. By Scheme

These subtopics are organized in relation with each other and with super topics by the design pattern, in terms of basic conceptual schema. The architecture of ontology built out this way captures explicitly the knowledge structure of the expertise of the domain.

5 Conclusion

The Topical Ontology of Frauds attempts to capture the knowledge structure and know-how of fraud investigators and prosecutors. It involves multiple disciplines. Its coverage changes with the dynamic evolution of the core concept, frauds. In order to manage the complexity and adapt to fast changes, emphasis is put on the architecture of ontology with an explicit specification of the knowledge structure of domain through the topical concepts and their configuration. The experience reported here is limited in that the extensive development of the topical ontology of fraud is needed to cover the full life cycle of frauds. It shows, however, the promise of added value for the ontology application, scalability, usability and descriptive flexibility by an architecture-oriented ontology engineering.

References

1. Albrecht, W.: Fraud Examination, Thomson South-Western, Ohio (2003)
2. Delgado, J. et al. An Ontology for Intellectual Property Rights: IPROnto. In Proceedings of the 1st International Semantic Web Conference (2002)
3. Halpin, T.: Information Modeling and Relational Databases: from Conceptual Analysis to Logical Design, Morgan Kaufmann, San Francisco (2001)
4. Jarrar, M., Demy, J., Meersman, R.: On Using Conceptual Data Modeling for Ontology Engineering. Journal on Data Semantics, LNCS, Springer (2003) 1238 – 1254
5. Masolo, C. et al: The WonderWeb library of Foundational Ontologies, Deliverable D17 Preliminary Report (2002)
6. Meersman, R.: Reusing Certain Database Design Principles, Methods and Techniques for Ontology Theory, Construction and Methodology, STARLab Technical Report, Vrije Universiteit Brussel (2000)
7. Niles, I., Pease, A: Towards a Standard Upper Ontology. Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (2001)
8. Wells, J.: Encyclopedia of Fraud, Association of Certified Fraud Examiners, Texas (2005)
9. Zhao, G., Kingston J., Kerremans K., Coppens F., Verlinden R., Temmerman R. & Meersman R.: Engineering an Ontology of Financial Securities Fraud. On the Move to Meaningful Internet Systems 2004: OTM Workshops, LNCS 3292, Springer Verlag (2004) 605 – 620
10. Zhao, G. & Meersman R.: Architecting Ontology for Scalability and Versatility. In, Meersman R., Tari Z. et al.,(eds.), *On the Move to Meaningful Internet Systems 2005: DOA, ODBASE and CoopIS, vol. 2*, LNCS 3761, Springer Verlag (2005) 1605 – 1164

Product Data Interoperability Based on Layered Reference Ontology

Wonchul Seo[1], Sunjae Lee[1], Kwangsoo Kim[1], Byung-In Kim[1], and Jae Yeol Lee[2]

[1] Dept. of Industrial & Management Engineering, Pohang University of Science and
Technology, San 31, Hyojadong, Namgu, Pohang, South Korea, 790-784
Phone: +82-54-279-2195
{wcseo, supersun, kskim, bkim}@postech.ac.kr
[2] Dept. of Industrial Engineering, Chonnam National University, Gwangju, South Korea
jaeyeol@chonnam.ac.kr

Abstract. In order to cope with the rapidly changing product development environment, manufacturing enterprises are forced to collaborate with each other through establishing a virtual organization. In collaboration, designated organizations work together for a mutual gain based on product data interoperability. However, product data interoperability is not fully facilitated due to the semantic inconsistency among product data models of enterprises. In order to overcome the semantic inconsistency problem, this paper proposes a reference ontology called Reference Domain Ontology (RDO) and a methodology for supporting product data interoperability with semantic consistency using RDO. RDO describes the semantics of product data model and metamodel for all application domains in a virtual organization. Using RDO, application domains in a virtual organization can easily understand product data models of others without model transformation. RDO is built by a hybrid approach of *top-down* using an upper ontology and *bottom-up* based on the merging of ontologies of application domains in a virtual organization.

1 Introduction

Manufacturing enterprises are faced with a rapidly changing product development environment demanding innovative products on time due to the customer-centric market conditions and the movement of business competition toward value chain to value chain. Thus, in order to quickly respond this product development environment and to meet various customers' needs, collaboration is necessary among manufacturing enterprises through establishing a virtual organization. A virtual organization is a temporal alliance of the enterprises that aims to share their core competences. In a virtual organization, designated enterprises work together for mutual gain and they can cope with the changing environment with agility based on the synchronization across a broad scope of manufacturing activities performed by multiple participant enterprises [8]. To support collaboration, it is important to achieve product data interoperability among participant enterprises because it enables the participants to exchange product data in real-time, to respond to a turbulent market environment without delay and ul-

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 573 – 587, 2006.
© Springer-Verlag Berlin Heidelberg 2006

timately to support a various decision-making for gaining a mutual goal of the virtual organization. Participant enterprises in a virtual organization should be able to understand shared product data accurately and completely for achieving interoperability. Ontology can be used to increase the understandability of product data models [9]. Gruber [4] defines ontology as follows:

"Ontology is a formal explicit specification of a shared conceptualization."

Each application domain can formally describe the semantics of its product data using defined ontology. Since ontology precisely defines the associated meaning of the product data model of a certain domain in a virtual organization, other domains are able to understand the model by referring the ontology defined in the domain. In order to make ontology referable each other, it is necessary to integrate ontologies of all the application domains in a virtual organization.

Ontology can be grouped into three broad categories of upper, mid-level and domain ontology by level of abstraction [19]. An upper ontology is high-level, general, and defines the semantics of domain independent concepts. Lower level ontologies are built by using and extending the concepts of an upper ontology. Thus, a well established upper ontology can be reused by many domain ontologies allowing one to take advantage of the semantic richness of the relevant concepts already built into the ontology [19].

In this paper, we propose a methodology of building a reference ontology in order to improve product data interoperability among application domains and to support collaboration in a virtual organization. The reference ontology will be built by a hybrid approach of *top-down* by the ontology layering and *bottom-up* based on the integration of domain ontologies. The ontology layering by level of abstraction promotes fast building of the reference ontology from the domain ontologies, since it supports the integration of domain ontologies based on the relationship between the domain concepts and the reused upper concepts. However, the relationships are definitely partial because there are many concepts in domain ontologies which are not related to the reused upper concepts. Thus, in order to complement the partial relationships, the remainder of the domain ontology is merged into the reference ontology using *bottom-up* approach by the domain experts. Similarly, the ontology layering partially supports the integration of a certain reference ontology with other reference ontologies. A reference ontology describes the semantics of product data models of all domains in a virtual organization, so each domain in the organization can semantically understand product data models of others using the reference ontology. The reference ontology is agile and temporal such that it is created with the formation of a virtual organization, copes with changes of the organization, and disappears with the vanishment of the organization.

This paper is organized as follows. Section 2 provides a description of related works and section 3 explains ontology-based product data modeling of application domains. A suggested methodology for achieving product data interoperability based on a reference ontology and an example using this methodology are presented in section 4. Conclusion of this paper and future works are discussed in section 5.

2 Related Works

In [7], the needs of achieving product data interoperability, system application interoperability and process interoperability for successful collaboration in product development are discussed. Product data interoperability is essential for application interoperability and process interoperability in a collaboration environment.

Various approaches for product data interoperability and sharing have focused on definitions of standards which provide neutral intermediate formats. This standard-based approach does not require multiple point-to-point translations, so it reduces the number of translations among interacting domains. The International Organization of Standardization (ISO) 10303, known as STandard for the Exchange of Product model data (STEP), is an international effort for data exchange of solid models. It furnishes a single neutral format to enable one to share product data across the entire lifecycle of the product development. It supports the exchange of product data by 2-stage translation of source-to-neutral and neutral-to-target [16]. However, the exchange of product shape and shape-related information is the primal focus of STEP and it does not attempt to exchange the semantic meaning associated with the product or design, and so this associated meaning such as designer's intent can be lost through the exchange [14]. Therefore it does not fully support the semantic consistency of product data model through the entire product lifecycle. For example, when a certain domain possesses some associated information which is separated from the product data but is useful in product designing, it will not be transferred to other domains if STEP is used as a neutral format.

STEP categorizes various types of product data using Application Protocols (APs). Each AP is applicable to one or more lifecycle stages of a particular product [16]. In order to promote interoperability between such various STEP APs in the area of product data management, the STEP Product Data Management (PDM) Schema has been established as a result of a cooperation of PDES Inc. and ProSTEP [15]. It is a core set of entities in STEP for the exchange of a central, common subset of the data managed within PDM system. Though the STEP PDM schema provides the mapping of concepts for PDM, the semantic inconsistency problem still remains in the area of individual product data model with additional associated meaning.

In [14], the requirements for semantic interoperability among Computer-Aided Design (CAD) systems are presented and a neutral ontology which describes the semantics of concepts in all systems is built. Using this neutral ontology, each system can translate their domain concepts into others by 2-stages. This assumes that 1-to-1 mapping between domain concepts always exists. There are, however, frequent occasions when concepts cannot be mapped each other by 1-to-1 since each domain describes the semantics of concepts for its own purpose.

Product Lifecycle Management (PLM) Services provide a foundation for collaborative engineering as a result of eXtended Product Data Integration (XPDI) project of ProSTEP iViP association [2]. PLM Services 1.0 defines a STEP AP 214 compliant data model and is standardized by the Object Management Group (OMG). In alignment with the Model Driven Architecture (MDA) [13], the standard supports common working environment based on the communication on demand by accessibility to each PDM system. With this accessibility, a mechanism to grasp the meaning of product data model semantically is needed because PLM Services only provide the ability to access required product data when one needs. A methodology to make acquired product data through PLM Services understandable semantically should be proposed.

In [17], an approach to transform models based on ontology which describes the semantics of the metamodel is presented at conceptual level. A metamodel means the rendering of a language definition as an object model. The metamodel transformation is based on the transformation between ontologies, and a model is transformed by binding between metamodel and ontology. In [11], two major architectures, the heuristic-based approach and the sharable common ontology-based approach for mapping discovery between ontologies, are presented and some methods to implement above approaches are introduced.

In [19], ontologies are grouped into three broad categories by level of abstraction. An upper ontology contains basic and universal concepts, so it ensures generality and expressivity for a wide area of domains. There are a number of ongoing initiatives to define a standard upper ontology and some results of those initiatives are the Suggested Upper Merged Ontology (SUMO) and the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) ontology [19]. The development of SUMO whose purpose is to promote data interoperability, information search and retrieval, and automated inference is based on the merging of various existing upper ontologies [10]. The purpose of the DOLCE ontology is to enable effective cooperation among various agents and establishing consensus in a mixed society where agents cooperate with human beings [3]. If each application domain builds its ontology by using or extending an upper ontology, the relationship between concepts in domain ontologies and concepts in an upper ontology can be established. Note that this relationship, however, cannot fully support the integration between domain ontologies, because each domain can define new concepts regardless of the concepts in an upper ontology for its own purpose or freely extend the existing upper ontology concepts.

This paper proposes a methodology to overcome the problems of existing works, to support efficient collaboration, and to achieve product data interoperability. Although this methodology is not fully automated, it can be an initiative for product data interoperability using the integrated reference ontology based on the ontology layering.

3 Product Data Modeling Based on Ontology

We assume that product data modeling of each application domain is based on the 4-layer metamodeling architecture of MDA [13]. As a metamodel means the rendering of a language definition, the metamodeling approach will support the precise definition of the product data model and the achievement of interoperability with other product data models if the semantics of the metamodel and the product data model is formally described. Thus, each domain should define ontology to describe the semantics of its model and metamodel with modeling product data. A domain ontology is built based on an upper ontology to utilize the advantages of the ontology layering.

3.1 Metamodeling Architecture

In the metamodling architecture of MDA shown in Fig. 1, a model must be paired unambiguously with a definition of the modeling language syntax and semantics. And a metamodel is defined based on a standard metamodel definition language. This architecture provides an environment that every domain model can be compatible with each other.

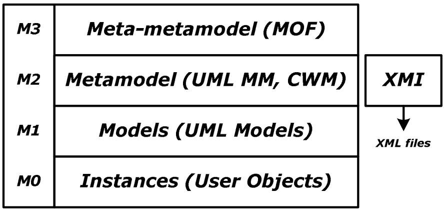
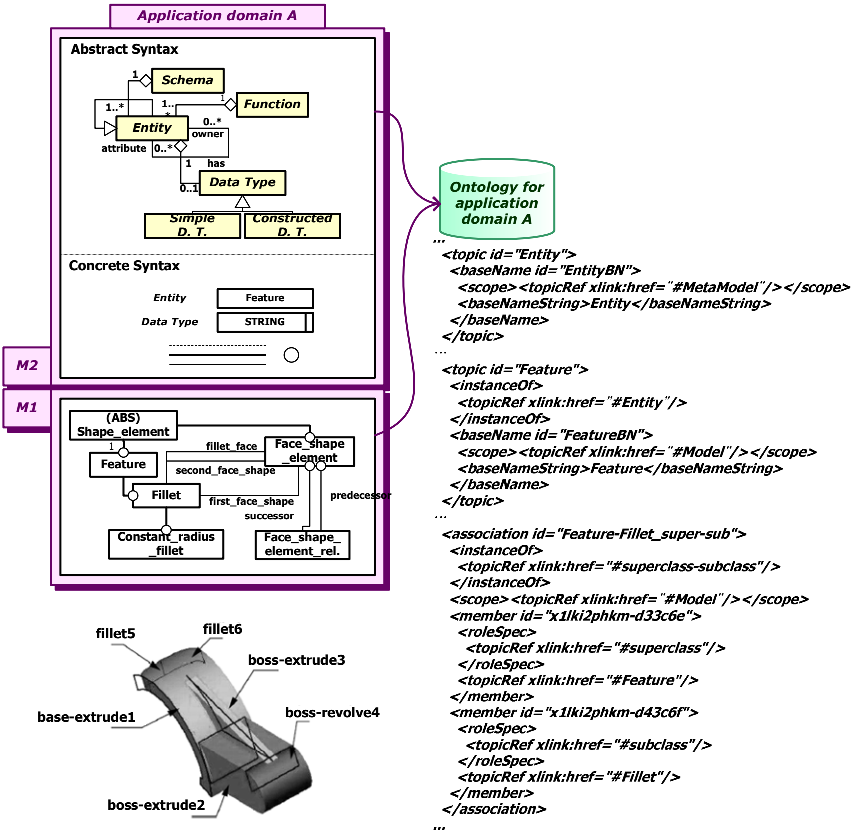


Fig. 1. MDA 4-layer metamodeling architecture

A meta-metamodel at the M3 layer defines the language structure for specifying a metamodel. The Meta Object Facility (MOF), an adopted OMG standard, provides a metadata management framework and a set of metadata services to enable the development and interoperability of model and metadata driven systems [12]. A metamodel at the M2 layer defines the language specification for a model in order to support the precise modeling, so it makes statements about what can be expressed in the valid models of a certain modeling language [18]. The purpose of modeling in every domain is different, and so modeling languages are different too. In the metamodeling architecture, Unified Modeling Language (UML) is a formal language for defining the structure and semantics of metadata and Common Warehouse Metamodel (CWM) is an adopted OMG standard for specifying the syntax and semantics of data in the warehouse domain [1]. A model at the M1 layer is a set of expressions about some systems using a certain modeling language. And XML Metadata Interchange (XMI) is a standard mechanism for interchanging metadata and metamodels in XML [1]. With the 4-layer metamodeling architecture of MDA, each application domain can unambiguously define its own product data model using language structure which is described in the metamodel.

3.2 Domain Ontology for Describing the Semantics of Product Data Model

A domain ontology formally describes concepts, relationships among concepts, and constraints which are used in the metamodel and model definition. For a simple example, a virtual application domain “A” is introduced. A subset of the metamodel and corresponding product data model is shown in Fig. 2. Example product data model and ontology shown in section 3 and 4 are built by referring works of [14] in order to compare a methodology and results of this paper with those of [14]. A domain “A” uses EXPRESS-G [5] for modeling its product data, so it describes a language definition of EXPRESS-G as a metamodel. The metamodeling architecture is different as a viewpoint for modeling is different. In the viewpoint of achieving interoperability among the product data models, a real thing is at M0 layer and a model for expressing the real thing according to the language definition described at M2 layer is at M1 layer. The domain builds ontology using Topic Maps [20] to describe formally the semantics of metamodel and product data model.



```
...
 <topic id="Entity">
  <baseName id="EntityBN">
   <scope><topicRef xlink:href=”#MetaModel”/></scope>
   <baseNameString>Entity</baseNameString>
  </baseName>
 </topic>
...
 <topic id="Feature">
  <instanceOf>
   <topicRef xlink:href=”#Entity”/>
  </instanceOf>
  <baseName id="FeatureBN">
   <scope><topicRef xlink:href=”#Model”/></scope>
   <baseNameString>Feature</baseNameString>
  </baseName>
 </topic>
...
 <association id="Feature-Fillet_super-sub">
  <instanceOf>
   <topicRef xlink:href="#superclass-subclass"/>
  </instanceOf>
  <scope><topicRef xlink:href=”#Model”/></scope>
  <member id="x1lki2phkm-d33c6e">
   <roleSpec>
    <topicRef xlink:href="#superclass"/>
   </roleSpec>
   <topicRef xlink:href="#Feature"/>
  </member>
  <member id="x1lki2phkm-d43c6f">
   <roleSpec>
    <topicRef xlink:href="#subclass"/>
   </roleSpec>
   <topicRef xlink:href="#Fillet"/>
  </member>
 </association>
...
```

Fig. 2. A subset of domain ontology and product data modeling of a virtual domain “A” based on the MDA metamodling architecture

There are other languages for the ontology definition such as Web Ontology Language (OWL) [21]. Though those ontology definition languages are different as the purpose of ontology is different, they play a similar role in the perspective of formal descriptions of the semantics. A selection problem of a suitable ontology definition language is out of scope of this paper. In this paper, Topic Maps are selected for ontology definition since they provide useful functions. Although other languages like OWL also offer similar functions, it is easy to implement those functions using Topic Maps. For example, it is necessary to classify and manage concepts based on their usage at the metamodel level and the model level because the concepts are coexisted and mixed in a domain ontology. For this classification, *scope* of Topic Maps can be used. In Topic Maps, *baseName* of *topic*, *association*, and *occurrence* contain *scope* as one of the content models [20]. *Scope* specifies the extent of the validity of a *topic* characteristic assignment. *Topic* is a resource that acts as a proxy for some subject and *association* is a relationship between *topics*. *Occurrence* is any information that is specified as being relevant to a given subject. As shown in Fig. 2, *scope* of "Entity" *topic* has value "MetaModel", *scope* of "Feature" *topic* and "Feature-fillet" *association* have value "Model". It means that the subject described by "Entity" *topic* is defined and used in the metamoel level and the subject described by "Feature" *topic* and "Feature-Fillet" *association* are defined and used in the model level.

3.3 Building Domain Ontology Using an Upper Ontology Based on the Ontology Layering

Each application domain describes the semantics of concepts for its own purpose, so many concepts may be interpreted differently if those belong to different domains. An upper ontology, however, defines the semantics of generic and domain independent concepts. Thus it can be reused by many domain ontologies. In order to utilize the advantage of the semantic richness, domain ontologies are built using an upper ontology as depicted in Fig. 3.

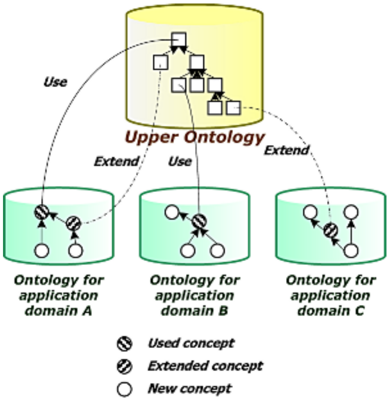

Use
Upper Ontology
Extend
Use
Extend
Ontology for application domain A
Ontology for application domain B
Ontology for application domain C
Used concept
Extended concept
New concept

Fig. 3. Building domain ontologies based on the relationship with an upper ontology

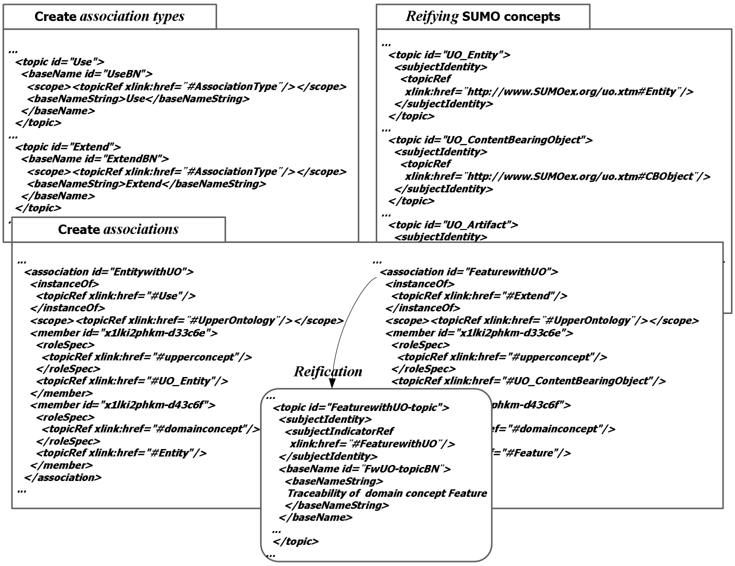
All domains use or extend the relevant concepts of the upper ontology in order to build their own domain ontology. Moreover, they may define new concepts without referring the upper ontology for their own needs. Such relationships between concepts in the domain ontology and the upper ontology should be clarified to integrate domain ontologies. So we assume that the upper ontology is described using Topic Maps too. With this assumption, the relationship can be expressed using *association* in Topic Maps and be implemented by the steps as follows:

Step 1: Specifying the relationship between concepts in the domain ontology and the upper ontology

Step 2: Creating an *association type* corresponding with a specified relationship unless the *association type* already exists

Step 3: Instantiating an *association* representing the specified relationship from the pertinent *association type*

It is possible to give an additional meaning into *association* through *reification* in Topic Maps. The meaning about a relationship between the domain ontology and the upper ontology can be managed semantically through reifying the *association*. Newly created *association types* and *associations* are also classified from the semantics of the metamodel and the model by using *scope*.



```
...
 <topic id="Use">
  <baseName id="UseBN">
   <scope><topicRef xlink:href=”#AssociationType”/></scope>
   <baseNameString>Use</baseNameString>
  </baseName>
 </topic>
...
 <topic id="Extend">
  <baseName id="ExtendBN">
   <scope><topicRef xlink:href=”#AssociationType”/></scope>
   <baseNameString>Extend</baseNameString>
  </baseName>
 </topic>
```

```
...
 <topic id="UO_Entity">
  <subjectIdentity>
   <topicRef
    xlink:href=”http://www.SUMOex.org/uo.xtm#Entity”/>
  </subjectIdentity>
 </topic>
...
 <topic id="UO_ContentBearingObject">
  <subjectIdentity>
   <topicRef
    xlink:href=”http://www.SUMOex.org/uo.xtm#CBObject”/>
  </subjectIdentity>
 </topic>
...
 <topic id="UO_Artifact">
  <subjectIdentity>
```

```
...
 <association id="EntitywithUO">
  <instanceOf>
   <topicRef xlink:href="#Use"/>
  </instanceOf>
  <scope><topicRef xlink:href=”#UpperOntology”/></scope>
  <member id="x1lki2phkm-d33c6e">
   <roleSpec>
    <topicRef xlink:href="#upperconcept"/>
   </roleSpec>
   <topicRef xlink:href="#UO_Entity"/>
  </member>
  <member id="x1lki2phkm-d43c6f">
   <roleSpec>
    <topicRef xlink:href="#domainconcept"/>
   </roleSpec>
   <topicRef xlink:href="#Entity"/>
  </member>
 </association>
...
```

```
...
 <association id="FeaturewithUO">
  <instanceOf>
   <topicRef xlink:href="#Extend"/>
  </instanceOf>
  <scope><topicRef xlink:href=”#UpperOntology”/></scope>
  <member id="x1lki2phkm-d33c6e">
   <roleSpec>
    <topicRef xlink:href="#upperconcept"/>
   </roleSpec>
   <topicRef xlink:href="#UO_ContentBearingObject"/>
phkm-d43c6f">
ef="#domainconcept"/>
f="#Feature"/>
```

```
...
 <topic id="FeaturewithUO-topic">
  <subjectIdentity>
   <subjectIndicatorRef
    xlink:href=”#FeaturewithUO”/>
  </subjectIdentity>
  <baseName id=”FwUO-topicBN”>
   <baseNameString>
    Traceability of  domain concept Feature
   </baseNameString>
  </baseName>
 ...
 </topic>
...
```

Fig. 4. A subset of ontology of domain “A” using the SUMO as an upper ontology

A subset of ontology of domain “A” shown in Fig. 4 is built using the SUMO as an upper ontology. As following the above steps, required *association types* are created. And concepts in the SUMO which the domain ontology refers are reified in order to use those concepts in the domain ontology. Through the *reification*, *topics* are created in the domain ontology from *topics* in the SUMO. For the simplicity of the example, only concepts are considered. With given *association types* and reified *topics*, *associations* are instantiated and they can have an additional meaning through *reification*.

4 Semantic Interoperability of Product Data Based on Reference Domain Ontology

The integration of domain ontologies in a virtual organization is important for product data interoperability because the semantics of product data of a domain is described by the domain ontology. We propose a Reference Domain Ontology (RDO) and a methodology using it for the integration of domain ontologies and the product data interoperability. The methodology enables participant domains to understand the product data model of others by referring RDO. Since RDO is built by merging all the domain ontologies in a virtual organization, it can describe the semantics of all the application domains. Thus application domains are able to understand the product data model of others by referring RDO and ultimately they can achieve product data interoperability. RDO is also implemented using Topic Maps for the compatibility with an upper ontology and domain ontologies. In this paper, the term "merge" is used in a broad sense, not limiting the meaning as different things are combined into one whole thing. Its meaning includes integration.

4.1 Building Reference Domain Ontology

The building process of RDO is depicted in Fig. 5 conceptually. First, RDO is created by a coordinator domain ontology in a virtual organization. Then it is extended by merging all the domain ontologies. Merging can be done by *top-down* and *bottom-up* approaches as described below.

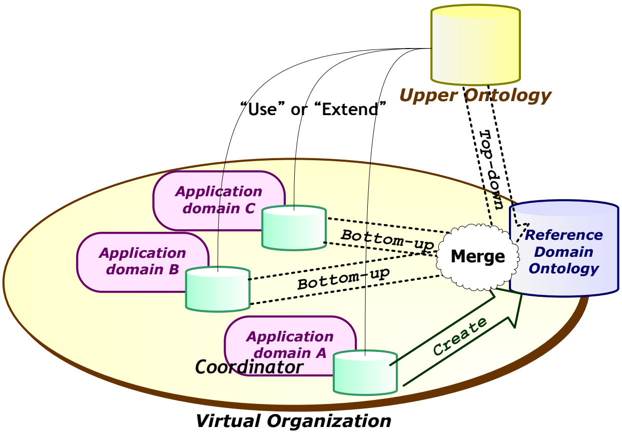


Fig. 5. Building RDO by *top-down* and *bottom-up* for a virtual organization

- *Top-down* approach

Concepts of domain ontologies that are derived from the concepts of the upper ontology by using "use" or "extend" can be merged into RDO by referring the relationship between the upper ontology and the domain ontologies as explained in section 3.3. This *top-down* approach prevents duplication of generic concepts and partially supports the integration with RDOs of other virtual organizations.

- *Bottom-up* approach

Domain ontologies of participant application domains in a virtual organization are merged into RDO. In order to prevent redundant merging, the semantics-based search should precede the actual merging. Each application domain ontology can be merged into RDO selectively and progressively based on the search. We consider Topic Maps Query Language (TMQL) as a semantics search language. TMQL is a standardization project of ISO and International Electrotechnical Commission (IEC). Although it is at working draft stage, it provides formal language to support semantic search from Topic Maps based data [6]. In order to achieve the coherence on all the concepts in RDO and to manage RDO efficiently, each concept in RDO specifies which domain it belongs to. This coherency is also implemented using *scope*. As scope enables one to trace the domain which possesses certain concepts, RDO is able to actively respond to rapidly changing domain ontologies.

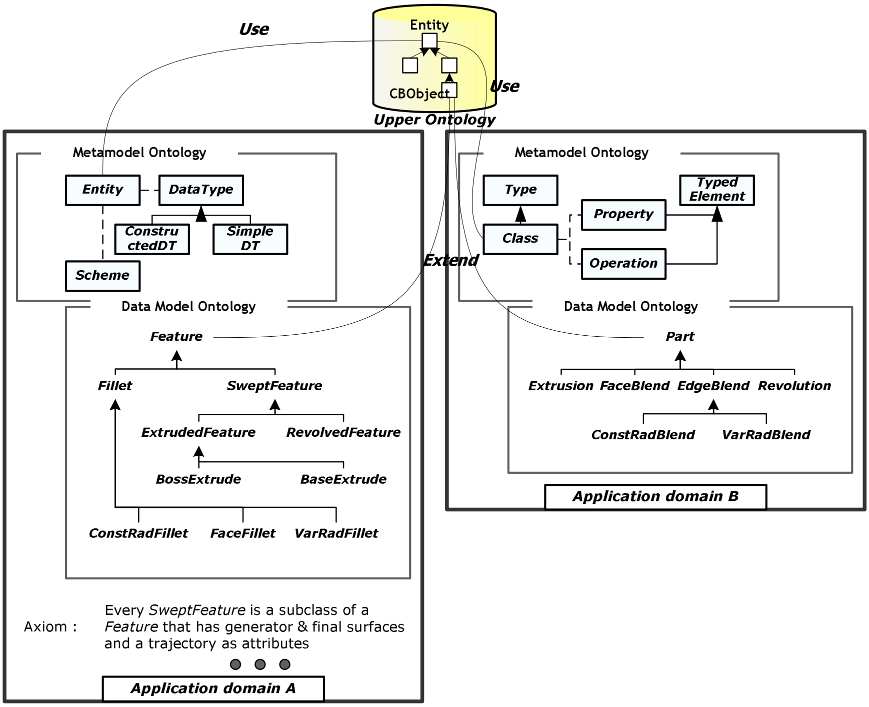
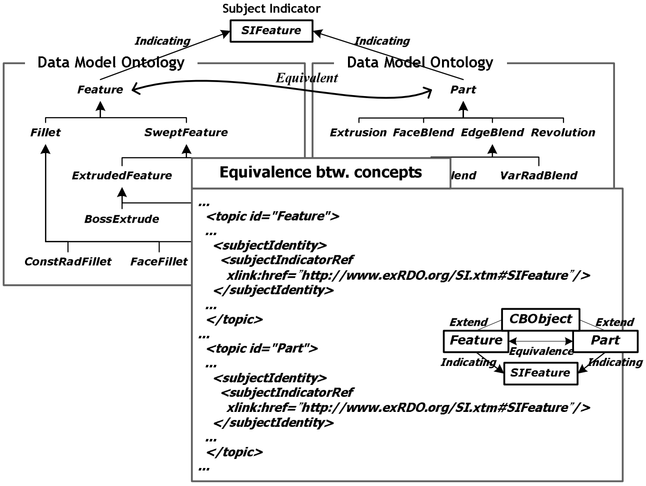


Fig. 6. Subsets of ontologies of virtual domain “A” and “B”

Subsets of ontologies of domain “A” and “B” with partial relationships with an upper ontology are shown in Fig. 6. Application domains “A” and “B” want to collaborate through establishing a virtual organization. For the simplicity of the example, only concepts which are defined and used in each domain and their hierarchical structures are displayed. First, RDO is initially created from ontology of domain “A” which is assumed be a coordinator in the virtual organization. Afterward, ontology of domain “B” is merged into RDO using *top-down* and *bottom-up* approaches by the domain experts of “B”. Topic Maps furnish basic merge and duplicate suppression operations for *topics* and *associations* based on *subject indicator*, *role players* of *association*, and etc. However, since these operations are simple characteristics-based, such as *baseName* or *subject indicator*, these cannot fully support the merging among different domain ontologies which requires the semantics-based equivalence checking. These operations present preconditions and postconditions for operations to be executed automatically. Thus these operations can be used to adjust RDO, even though these are not suitable for the full merging. For example, if a *topic* is determined equivalent to another already defined in RDO, it will be modified to indicate the same subject and two *topics* will be merged through basic operations in Topic Maps.



```
...
 <topic id="Feature">
 ...
  <subjectIdentity>
   <subjectIndicatorRef
    xlink:href="http://www.exRDO.org/SI.xtm#SIFeature"/>
  </subjectIdentity>
 ...
 </topic>
...
 <topic id="Part">
 ...
  <subjectIdentity>
   <subjectIndicatorRef
    xlink:href="http://www.exRDO.org/SI.xtm#SIFeature"/>
  </subjectIdentity>
 ...
 </topic>
...
```

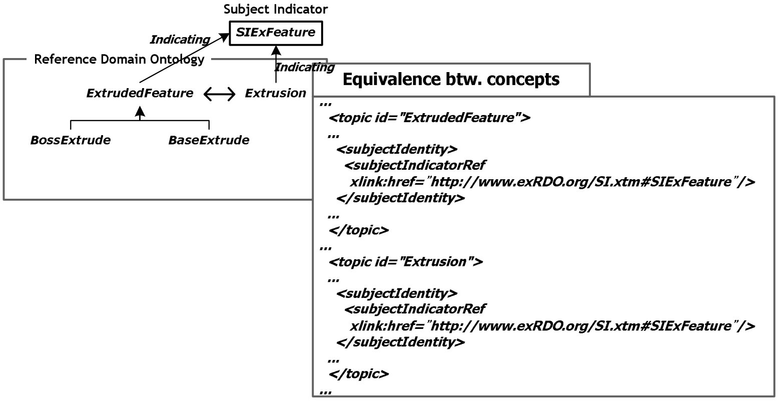
Fig. 7. Merging "Part" *topic* into RDO based on the upper ontology

- Example of *top-down* approach

Domain experts of "B" evaluate the relationship between concepts in RDO and in domain "B". In the *top-down* approach, only the concepts built based on the upper ontology are evaluated. In Fig. 6, "Entity" in domain "A" initially inserted into RDO and "Class" in domain "B" both use "Entity" of the upper ontology. The two concepts can be determined as equivalent and be merged because they use the same upper ontology concept "Entity". And "Feature" in RDO and "Part" in domain "B" extend "CBObject" of the upper ontology. The evaluation between the two concepts is performed based on the semantic information in *associations* which define the relationship between the domain concepts and the upper concepts. "Feature-CBObject" *association* in domain "A" describes specifically how "Feature" extends "CBObject" and "Part-CBObject" *association* in domain "B" describes how "Part" extends "CBObject" by *reification* as mentioned in section 3.3. With this evaluation, "Part" is merged into RDO. In order to support the semantics-based evaluation, some useful services for RDO such as a graphic-based navigation service and a TMQL query service can be provided to domain experts of "B" who are merging their domain ontology into RDO. The result of merging "Part" is partially depicted in Fig. 7. In order to describe the equivalence, a *subject indicator* for new subject is created in RDO and equivalent *topics* set this *indicator* as *subjectIdentity*. Now the merging process is completed and equivalent *topics* will be merged by basic operations in Topic Maps. If the concepts in domain “B” cannot be related with the concepts in RDO based on the upper ontology, they are merged by the *bottom-up* approach. Concepts that do not refer the upper ontology are also merged into RDO by the *bottom-up* approach.

- Example of *bottom-up* approach

In the *bottom-up* approach, remained concepts in domain “B” after executing the *top-down* approach are merged. The relationship between the concepts is categorized into three distinct cases: 1-to-1 equivalent relationship, 1-to-many or many-to-1 hierarchical relationship, and no relationship. If there is a 1-to-1 equivalent relationship between “ExtrudedFeature” in RDO and “Extrusion” in domain “B”, they will be merged in the same way with the *top-down* approach shown in Fig. 8. Other remaining concepts that have not 1-to-1 relationship should be merged into RDO with correct structural meaning. Fig. 9 shows various structural patterns that “Extrusion” in domain “B” can be merged into RDO.



```
...
 <topic id="ExtrudedFeature">
 ...
   <subjectIdentity>
     <subjectIndicatorRef
       xlink:href="http://www.exRDO.org/SI.xtm#SIExFeature"/>
   </subjectIdentity>
 ...
 </topic>
...
 <topic id="Extrusion">
 ...
   <subjectIdentity>
     <subjectIndicatorRef
       xlink:href="http://www.exRDO.org/SI.xtm#SIExFeature"/>
   </subjectIdentity>
 ...
 </topic>
...
```

Fig. 8. Merging “Extrusion” *topic* into RDO in the case of 1-to-1 equivalent relationship

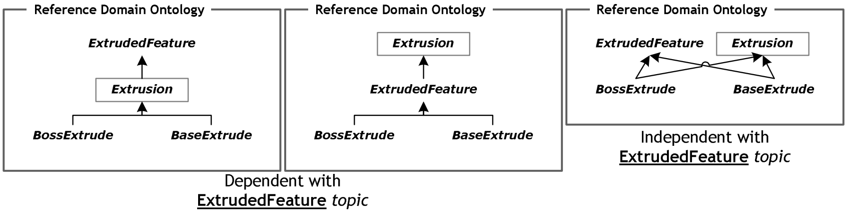


Fig. 9. Various structural patterns for merging of “Extrusion” *topic* into RDO

RDO is built by a hybrid approach of *top-down* and *bottom-up*. A subset of an example RDO which is built by virtual domain “A” and “B” is shown in Fig. 10. This example RDO has only a structural model of the concepts for simplicity, but RDO should have relationships between concepts, constraints, axioms, and associated information such as designer’s intent.

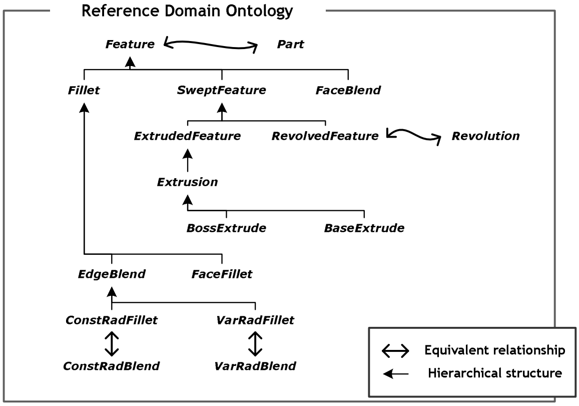


Fig. 10. A subset of an example RDO by merging domain ontologies of virtual domains

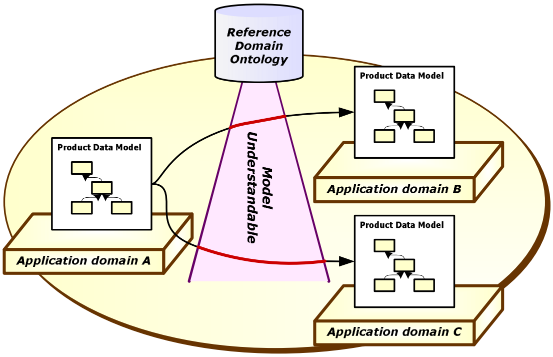


Fig. 11. Product data interoperability in a virtual organization based on the model understanding using RDO

4.2 Product Data Interoperability in a Virtual Organization Using Reference Domain Ontology

In a virtual organization, a coordinator creates RDO using its own domain ontology and then designated enterprises participating in the organization merge their ontology into RDO. Participant enterprises are able to achieve the semantic consistency about the product data model of others using RDO, since RDO formally describes the semantics of all the domains. Participants can understand syntactically and semantically the shared product data model without model transformation as shown in Fig. 11.

The methodology presented in this paper using RDO built by a hybrid approach of *top-down* and *bottom-up* provides following advantages : 1) Coherence with RDOs of other virtual organizations based on the ontology layering, 2) Increasing agility of the building process of RDO and ultimately of the virtual organization through reusing upper ontology, 3) Achievement of the consistency about product data models of participant enterprises syntactically and semantically, 4) Fast response to the rapidly changing environment of the virtual organization. The methodology does not require the 1-to-1 transformation of each product data model. Thus it can support to achieve product data interoperability accurately and completely based on the integrated semantics. The building process of RDO, however, is not fully automated. This may hinder the agile collaboration, since the users working collaboratively should do it in everything. Although the full automation is valuable, it is very hard to implement. So there are various researches for semi-automation based on the semantics similarity, the evaluation of the ontology users, and so on. Thus these valuable existing researches will be able to boost the ability of this work to support the agile collaboration in a virtual organization based on product data interoperability.

5 Conclusion

In order to agilely respond the rapidly changing product development environment, manufacturing enterprises are forced to establish a virtual organization sharing their own core competences. The collaboration should be based on product data interoperability among participant enterprises to support a successful decision-making for mutual gain without delay.

In this paper, RDO as a reference ontology for the virtual organization and a building methodology of RDO is presented to achieve product data interoperability. RDO can be built by a hybrid approach of *top-down* using an upper ontology and *bottom-up* by the integrated semantics merging ontologies of all the participant domains. Using RDO, participant domains are able to acquire the ability to understand the product data of others in the virtual organization and ultimately achieve product data interoperability within the organization, since RDO formally describes the semantics of the metamodel and the product data model of all the domains. Although the process for merging domain ontologies is not fully automated, this methodology can be an initiative for product data interoperability based on the ontology layering. For future works, the mechanism for automatic merging of domain ontologies should be addressed. Moreover, participant domain ontologies tend to be changed rapidly as time passes. In order to cope with the rapidly changing domain ontologies, the evolution management framework for RDO should be presented. And for application system interoperability, development of a framework for integrating PDM systems using PLM Services and the semantics of product data based on Web Services can be pursued.

Acknowledgement

This work is supported by grant no. R01-2003-000-10171-0 from the Basic Research Program of the Korea Science and Engineering Foundation.

References

1. Chang, D.: Common Warehouse Metamodel (CWM), UML and XML. presentation to the Metadata Conference/DAMA Symposium. Arlington, VA. March 22 (2000) Online available at http://www.cwmforum.org/cwm.pdf
2. Feltes, M.: PLM Services - a Standard to Implement Collaborative Engineering. Daimler Chrysler Research (2005) Online available at http://www.prostep.org/file/17050.intro

3. Gangemi, A., Guarino, N., Masolo, C. and Oltramari, A.: Sweetening wordnet with DOLCE. AI Magazine, Vol.24, No.3 (2003) 13-24
4. Gruber, T. R.: A translation approach to portable ontology specifications. Knowledge Acquisition, Vol.5, No.2 (1993) 199-220
5. ISO: ISO 10303 Industrial automation systems and integration - Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual. December 15 (1994)
6. ISO: ISO 18048 ISO/IEC JTC1/SC34 Information Technology — Document Description and Processing Languages: Topic Maps Query Language (TMQL). February 18 (2005) Online available at http://www.isotopicmaps.org/tmql/spec.html
7. Kim, H., Kim, H.-S., Lee J.-H., Jung, J.-M., Lee, J. Y. and Do, N.-C.: A framework for sharing product information across enterprises. The International Journal of Advanced Manufacturing Technology, Vol.27, No.5-6 January (2006) 610-618
8. MESA: Collaborative Manufacturing Explained. A MESA International White Paper (2004) Online available at http://www.mesa.org
9. Mizoguchi, R. and Ikeda, M.: Towards Ontology Engineering. Technical Report AI-TR-96-1, I.S.I.R. Osaka University (1996)
10. Nichols, D. and Terry, A.: User's Guide to Teknowledge Ontologies. Teknowledge Corp. December 3 (2003) Online available at http://ontology.teknowledge.com/Ontology_User_Guide.doc
11. Noy, N. F.: Semantic Integration: A Survey Of Ontology-Based Approaches. Special Interest Group on Management Of Data (SIGMOD) Record, Vol.33, No.4 December (2004) 65-70
12. OMG: MOF Core Specification. OMG available specification v2.0. January (2006)
13. OMG: MDA. OMG Document number ormsc/2001-07-01. July (2001)
14. Patil, L., Dutta, D. and Sriram, R.: Ontology-Based Exchange of Product Data Semantics. IEEE Transactions on Automation Science and Engineering, Vol.2, No.3 July (2005) 213-225
15. PDM Implementor Forum: Usage Guide for the STEP PDM Schema (Release 4.3). January (2002) Online available at http://www.pdm-if.org/pdm_schema/pdmug_release4_3.zip
16. Pratt, M. J.: Introduction to ISO 10303 - the STEP Standard for Product Data Exchange. Journal of Computing and Information Science in Engineering, Vol.1, No.1 March (2001) 102-103
17. Roser, S. and Bauer, B.: Ontology-based Model Transformation. Proceedings of the ACM/IEEE 8th International Conference On Model Driven Engineering Languages And Systems (MoDELS/UML-2005) - Posters. Montego Bay, Jamaica (2005)
18. Seidewitz, E.: What Models Mean. IEEE Software, Vol.20, No.5 (2003) 26-32
19. Semy, S. K., Pulvermacher, M. K. and Obrst, L. J.: Toward the Use of an Upper Ontology for U.S. Government and U.S. Military Domains: An Evaluation. MITRE Technical Report. (2004) Online available at http://www.mitre.org/work/tech_papers/tech_papers_04/04_0603/04_1175.pdf
20. TopicMaps.org: XML Topic Maps(XTM) 1.0. August 6 (2001) Online available at http://www.topicmaps.org/xtm/1.0/
21. W3C: OWL Web Ontology Language Overview. W3C Recommendation. February 10 (2004) Online available at http://www.w3.org/TR/owl-features/

Design of Semantically Interoperable Adverse Event Reporting Framework*

Senator Jeong and Hong-Gee Kim

DERI Seoul, Seoul National University
Yeongeon-Dong 28, Joengno-Gu, Seoul, Korea
{senator, hgkim}@snu.ac.kr

Abstract. Patient safety is one of the most significant issues not only to medical providers but also to the general public. Despite the widespread recognition of the adverse event reporting for patient's safety, there is no widely accepted or standardized way to request and report adverse event information. We designed the semantically interoperable Adverse Event Reporting framework. It consists of two components: the Adverse Event Ontology to describe adverse event in semantically interoperable way and the Adverse Event Reporting Schema (AERS) to envelope and deliver the content of adverse event report request and report. The Adverse Event Ontology was built upon existing adverse event taxonomies. The AERS was designed for common adverse event messaging interface in the form of XML Schema. The adverse event reporting framework is expected to allow semantic interoperability in sharing and exchange of patient safety information within and among various healthcare information systems.

1 Introduction

Patient safety is one of the most significant issues not only to medical providers but also to the general public in many aspects of healthcare because *adverse events* threatening patient safety occur frequently and even trivial often result in severe harm. Adverse event is any event that we do not wish to have happened again[1]. The notion of *adverse event reporting* is that when a reportable adverse event occurs, then it should be reported to the designated recipients. The purpose of adverse event reporting is to understand their origin, predict their occurrence, draw out corrective and preventive actions, and implement quality improvement strategies[2]. There are numerous adverse event reporting systems for specific information need. They collect data on medication errors[3-5], adverse events involving medical products[6], reactions[7, 8], or data solely at specific domain or organization[9].

Despite many reporting systems have been implemented, the ability to learn from these systems is limited because they do not ***talk*** to each other. Data are not combined or aggregated in the same manner because there is no standardized system for

[*] This study was supported by a grant of the interoperable EHR Research and Development (A050909), Ministry of Health & Welfare, Republic of Korea.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 588 – 594, 2006.

© Springer-Verlag Berlin Heidelberg 2006

classifying and categorizing patient safety problems[10]. The terminologies meaning adverse event vary and nomenclatures are discordant among vocabularies. And it makes difficult to share and exchange adverse event information among different healthcare information systems. In addition, no global standard provides formal messaging format for adverse event reporting. The methods used to record adverse events vary among report requesters, aggregators, and investigators.

The main goal of our study is to provide semantically interoperable adverse event reporting framework. To this end, we built the Adverse Event Ontology, which provides a mechanism to resolve coding disagreement between healthcare agents. Then, we designed the Adverse Event Reporting Schema, which will be used to represent common message interface between adverse event report requesters and reporters. Next, since different principals may have different information needs we designed the Report Item Sets, which function as report item templates for specific user's preference and domain. Finally, we developed a prototype system to demonstrate the proposed framework.

2 Adverse Event Ontology

One of our goals is to develop ontology with logical construction across multiple domains to the detailed level and to capture as many different event types as possible.

The Adverse Event Ontology was built upon earlier patient safety taxonomy research conducted by previous works[11-16] and extended them into a more comprehensive ontology. We modeled the ontology in OWL DL plug-in. the ontology has five high level primary classifications as in [11, 12]: Impact, Type, Domain, Cause, and Prevention & Mitigation. Impact is the outcome or effects of medical error and systems failure commonly referred to as harm to the patient. Type is the implied or visible processes those were faulty or failed. Domain is the characteristics of the setting in which an incident occurred and the type of individuals involved. Cause is the factors and agents that led to an incident. Prevention & Mitigation is the measures taken or proposed to reduce incidence and effects of adverse occurrences. The Ontology has several properties and disjoints. The properties of the ontology include for example *hasInput*, *hasLevel*, *hasCheck,* and something like that. The structure of the ontology is as Fig.1. and the schema is available at *http://chord.snu.ac.kr/~senator/safety/pseo.owl.*

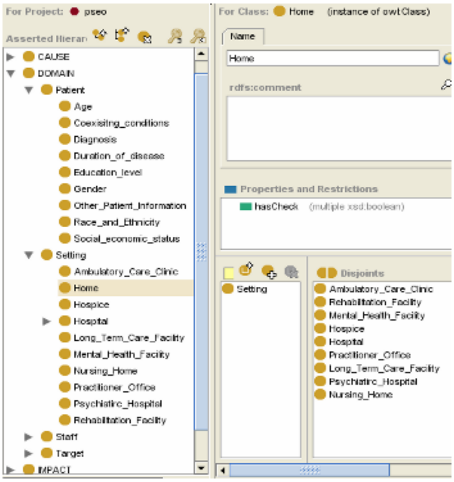


Fig. 1. The Adverse Event Ontology

3 Adverse Event Reporting Schema

The very concept of event reporting is that when a specific event occurs at the predefined condition, then it is reported to relevant recipient. As in case of adverse event data, reporting forms differ depending on report requester, aggregators, and investigator in the sense that there is no commonly usable report data interchange interface. To date, however, attempts have been hardly made to build standardized messaging interface across institution boundary. We need a means to exchange information about adverse events between various healthcare principals. Therefore a unified messaging interface for all types of adverse event reporting would be highly desirable. Considering this need we designed the XML based Adverse Event Reporting Schema (AERS). The AERS is intended to become a common messaging tool used by healthcare consumers, providers, regulators, or other principals when they describe whatever they want.

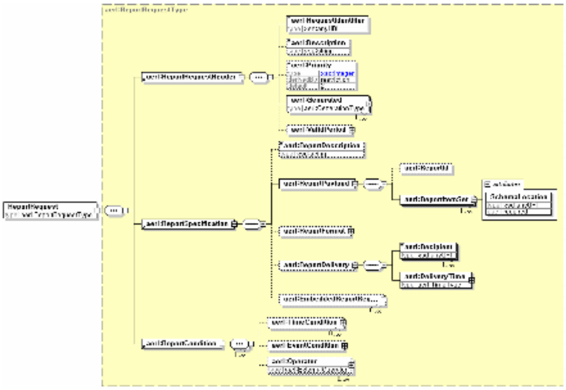


Fig. 2. XML Schema for Adverse Event Report Request

ReportRequest. The AERS comprises of `ReportRequest` and `Report`. The purpose of `Report Request` is to describe the adverse event, which is asked to be reported, designate the recipient, and specify a Report Item Set by which report data payload is included in a report. The `ReportRequest` is composed of three main sections consisting of several parts as illustrated in Fig. 2. In the `ReportRequestHeader` the `Priority` specifies the priority level (0 to 5) for a Report Request to be processed by the system. The `ValidPeriod` defines the life time of a Report Request. The `ReportSpecification` allows Report Requesters to specify which report items should be included in report payload, who is its recipient, and when it is delivered. For instance, a Requester can specify an xml schema location of Report Item Set which would be imported by Report Generator. The `DeliveryTime` in the `ReportDelivery` allows requestor to specify the time a report is delivered. Using `ReportCondition` requesters are able to specify report conditions under which Reports are reported: adverse event type, time-span events occur, or combinations thereof.

Report. The Report schema has three main elements. As in Report Request the `ReportHeader` is used to provide general descriptions of Report. The `ReportItemSet` provides a place for inclusion of report's payload. It corresponds to the `ReportItemSet` that is specified in the originating Report Request. The optional element `EmbeddedReportRequest` contains embedded Event Report Request or reference thereof.

Report Item Set. Information needs may differ depending on communication parties (e.g. healthcare provider, trading partner, patient), communication scope (within or cross organization), healthcare setting (e.g. hospital, ambulatory care setting, home care etc.), and reporting type (e.g. accountability reporting, ad hoc reporting). Some users want simple report data while others want more details applicable to their business domain. Hence, report requesters should be given a set of options to choose a Report Item Set which is deemed to be most qualified to satisfy their information needs. The main purpose of the Report Item Set is to function as reported data template which is filled in by Reporter. Stakeholders might extract report items from the Adverse Event Ontology and build a Report Item Set Schema with help of Report Item Set Generator as shown in Fig.6. For demonstration we designed an exemplary Report Item Set as shown in Fig. 3.

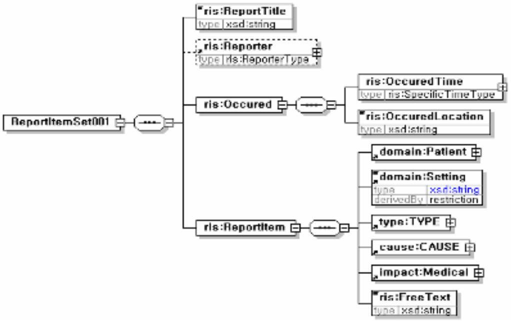


Fig. 3. Exemplary Report Item Set Schema

4 The Adverse Event Reporting System

In this section we describe the Adverse Event Reporting System and its use-case scenario. The system consists of four components; The *Report Requester* (public surveillance system manager, individual healthcare quality improvement manager, or agent thereof) who is able to access to the *Report Repository* through authentication; The *Report Generator* who is responsible for generating Report(s); The *Report Repository* which record the adverse event reports specified in a given report request; The *Report Item Set Library* which is referenced to generate Report Request(s) and Adverse Event Report(s). The Library provides Adverse Event Report Item Sets which are extracted from the Adverse Event Ontology and used to specify reported items in a report.

The reporting system operates as numbered sequence illustrated in Fig.4. Using Request Generator the Report Requester selects Report Item Set from the Report Item Set Library to generate a Report Request in which the *Report Time Condition* is set to '*any event occurred during 10 days from January 5, 2006*' and Event Condition is '*Death*'. In the Report Request two Report Recipients (*SH* | *PSE-RP-003*)[1] were designated. Then it is delivered to Request Recipients. A *Report Request* generated by the *Report Request Generator* is as Fig. 6. On receiving Report Request (*GH-RR-001*), the Reporter (*MH*) captures a '*Death*' event which had been gone through internal investigation procedures. Next, the Reporter generates an *Adverse Report* (MH-R-001) using the Report Generator which imports Report Item Set xml schema

[1] In this use-case, let's say RR is Report Request, R is Report, RP is Report Repository, GH is General Hospital, SH is Smart Hospital, MH is Marine Hospital, PSE-RP is Patient Safety Event.

into the Report payload specification and send it to two Recipients who are specified in the Report Request (GH-RR-001). The example of report generated by Report Generator is as Fig.7. The adverse event Repository (PSE-RP) is responsible for consolidating all information which will be offered to requesters. The repository is also used to gather accumulative adverse event statistics. The information may be total adverse events to date, types of events reported ever, and types of providers reporting. report requesters are able to search the repository to retrieve report data which they are interested in.

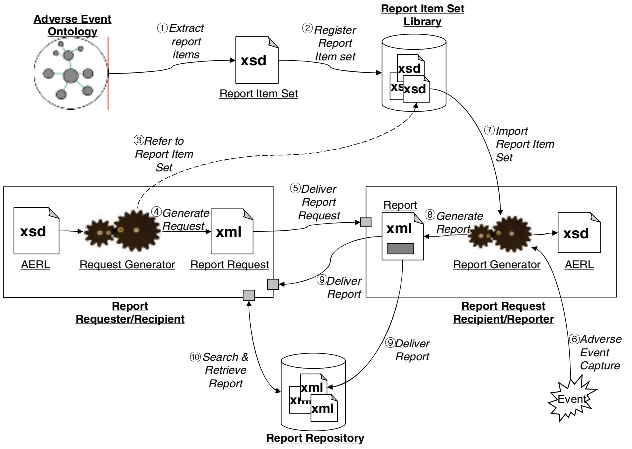
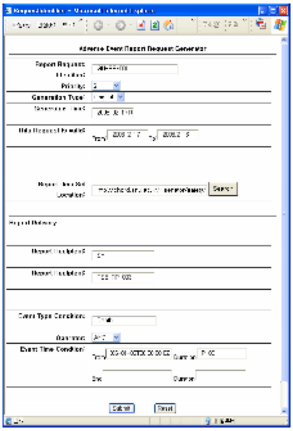


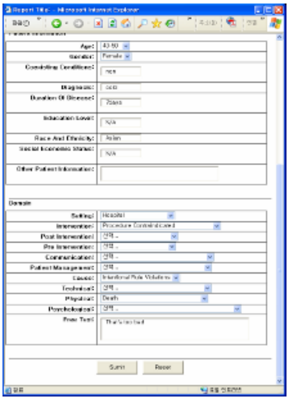
Fig. 4. Ontology-based Adverse Event Reporting System Architecture



Adverse Event Report Request Generator

Report Request Identifier:
Priority: 3
Generation Type:
Data Request Range: From 2006.2.1 To 2006.2.5
Report Item Set Location: Search
Report Delivery
Report Recipient:
Report Recipient:
Event Type Condition:
Event Time Condition: From Duration
End Duration

Fig.5. Report Request Generator's GUI

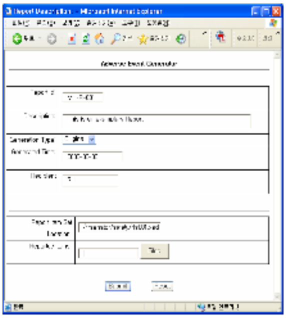


Age:	40-50
Gender:	Female
Coexisting Conditions:	none
Diagnosis:	[illegible]
Duration Of Disease:	[illegible]
Education Level:	N/A
Race And Ethnicity:	Asian
Social Economic Status:	N/A
Other Patient Information:	

Domain

Setting:	Hospital
Intervention:	Procedure Contraindicated
Post Intervention:	선택 -
Pre Intervention:	선택 -
Communication:	선택 -
Patient Management:	선택 -
Cause:	Intentional Rule Violations
Technical:	선택 -
Physical:	Death
Psychological:	선택 -
Free Text:	[illegible]

Fig.6. Report Items generator GUI



Adverse Event Generator

Generator Type

Fig.7. Report generator GUIZ

An adverse event report should be immediately disseminated to and shared among concerning parties so those who receive report could implement useful prevention strategies. Drastically simplifying the steps and reducing the time is required[1]. Considering these requirements, we designed the reporting system user interfaces so that users can input data entry as easily as possible. The Report Request Generator and Report Generator GUIs were built using XSLT. The system users are able to input data using these generation interfaces as in Fig.5-7.

5 Conclusions and Future Work

The purpose of adverse event reporting is to improve patient safety through greater sharing of information about adverse events. We proposed an ontology and xml schema driven methods for semantically interoperable adverse event data communication among geographically distributed and heterogeneous health care information systems.

This paper described the beginning stage of our work on the semantically interoperable adverse event reporting framework. Significant challenges remain to develop sound system to meet various information needs of adverse event reporting community. Above all things, field-test is required to determine plausibility and suitability of the proposed framework. Further, we've just built an exemplary Report Item Set schema by hand. The next stage of the project we will implement the engine which is able to semi-automatically extract elements from the Adverse Event Ontology to construct Report Item Sets depending on user's information need. Still another work to be done is deliberation method of message between Requester (Report Recipient) and Reporter (Request Recipient). In the next stage we explore efficient method for message delivery.

References

1. Fernald, D.H., Pace, W.D., Harris, D.M., West, D.R., Main, D.S., Westfall, J.M.: Event Reporting to a Primary Care Patient Safety Reporting System: A Report From the ASIPS Collaborative. Ann Fam Med 2 (2004) 327-332
2. Makeham, M.A.B., Dovey, S.M., County, M., Kidd, M.R.: An international taxonomy for errors in general practice: a pilot study. Medical Journal of Australia 177 (2002) 68-72
3. USPC: MedMarx system: THE NATIONAL DATABASE FOR MEDICATION ERRORS. Vol. 2006. U.S. Pharmacopeia
4. USP: Medication Errors Reporting Program. Vol. 2006. the Institute for Safe Medication Practices
5. United States. Food and Drug Administration.: Vaccine adverse event reporting system (VAERS) historic, Jan. 1, 1992 to Dec. 31, 1992. NTIS, Springfield, VA (1994) 2 computer disks
6. The Emergency Care Institute (ECRI). The Emergency Care Institute (ECRI)
7. Zhou, W., Pool, V., Iskander, J.K., English-Bullard, R., Ball, R., Wise, R.P., Haber, P., Pless, R.P., Mootrey, G., Ellenberg, S.S., Braun, M.M., Chen, R.T.: Surveillance for safety after immunization: Vaccine Adverse Event Reporting System (VAERS)--United States, 1991-2001. MMWR Surveill Summ 52 (2003) 1-24

8. BLAKE, M., PINKSTON, V.: Electronic Reporting of Adverse Event Data to the Food and Drug Administration--The Experiences of Glaxo Wellcome and Zeneca as Participants in the Adverse Event Reporting System Pilot Project. Drug Information Journal 33 (1999) 1101–1108
9. Mekhjian, H.S., Bentley, T.D., Ahmad, A., Marsh, G.: Development of a Web-based Event Reporting System in an Academic Environment. J Am Med Inform Assoc 11 (2004) 11-18
10. NQF: Standardizing a Patient Safety Taxonomy-tx Taxonomy Final for Web Public. A consensus report. National Quality Forum, Washington D.C (2006)
11. Chang, A., Schyve, P.M., Croteau, R.J., O'Leary, D.S., Loeb, J.M.: The JCAHO patient safety event taxonomy: a standardized terminology and classification schema for near misses and adverse events. International Journal of Quality in Health Care 17 (2005) 95-105
12. WHO: World Alliance for Patient Safety: Forward Programme. World Health Organization (2004)
13. Woods, D.M., Johnson, J., Holl, J.L., Mehra, M., Thomas, E.J., Ogata, E.S., Lannon, C.: Anatomy of a patient safety event: a pediatric patient safety taxonomy. Qual Saf Health Care 14 (2005) 422-427
14. Dovey, S.M., Meyers, D.S., Phillips, R.L., Jr., Green, L.A., Fryer, G.E., Galliher, J.M., Kappus, J., Grob, P.: A preliminary taxonomy of medical errors in family practice. Qual Saf Health Care 11 (2002) 233-238
15. Pace, W.D., Staton, E.W., Higgins, G.S., Main, D.S., West, D.R., Harris, D.M.: Database Design to Ensure Anonymous Study of Medical Errors: A Report from the ASIPS collaborative. J Am Med Inform Assoc 10 (2003) 531-540
16. Brixey, J., Johnson, T.R., Zhang, J.: Evaluation of a medical error taxonomy. The Annual Meeting of American Medical Informatics Association, San Antonio. (2002)
17. Keyser, V.D., Nyssen, A.-S., Lamy, M., Fagnart, J.-L., Baele, P.: Development of a critical incidents reporting system in medicine : final report. Development of a programme for reporting and analysing critical incidents in medical establishments Federal Science Policy, Brussels (2004)

Protein Data Sources Management Using Semantics

Amandeep S. Sidhu[1], Tharam S. Dillon[1], and Elizabeth Chang[2]

[1] Faculty of Information Technology, University of Technology, Sydney, Australia
{asidhu, tharam}@it.uts.edu.au
[2] School of Information Systems, Curtin University of Technical University, Perth, Australia
Elizabeth.Chang@cbs.curtin.edu.au

Abstract. Presently, organizations make significant investments in biomedical data and information sources. These investments are expected to produce reduction of errors and quality improvements in data management and analysis. To sustain achievements in quality and efficiency, healthcare organizations need to be vigilant in monitoring the state of competitiveness of their platforms. In the technology post-adoption period, healthcare organizations use multiple data sources to search for technology-related information to maintain technology parity with, or dominance over their competitors. Firstly this study seeks to answer the following research question: what approaches do healthcare organizations employ with regard to managing diverse sources of data and information in order to sustain their technology competitiveness. Then as an initial step in this direction, in this paper we discuss the conceptual foundation for the phenomenon of data and information sources management capability for the proteomics domain. This is done by discussing the case of Protein Data Source Integration by Protein Ontology.

Keywords: Protein Ontology, Biomedical Ontologies, Knowledge Management, Information Retrieval, Data Integration, Data Semantics.

1 Introduction

With increased affordability and mass proliferation of information systems, healthcare organizations have become more capable of aggregating information for treated individuals as well as for patient and disease populations. In technology-rich environments, a comprehensive protein information source, available online for researchers is highly salient. It aggregates various types of protein data and information such as entry details, structural information, functionality, chemical bonding details, genetic defects, etc. for individual protein complexes. The promise of providing a comprehensive protein information resource lies in improving protein identification, selection and protein-protein matching. Strategy selection in terms of managing integration between various protein data and information sources is therefore a critical investigation. The focus of this paper is to discuss our choice of strategy for managing protein data and information while we create Protein Ontology Framework [1, 2]. In particular, we want to draw a distinction between intra-organizational and cross-organizational-boundary integration of sources that we encountered while developing Protein Ontology [3, 4].

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 595 – 601, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Given diverse information needs of proteomics domain in the post-adoption period of technology use, system capability to manage various sources of information would be critical for preserving technology competitive advantage. To formulate the theoretical underpinnings for strategy selection for the common Protein Ontology Conceptual Framework for management of information sources, we turn to Resource-Based View (RBV), Dynamic Capabilities (DC) perspective and literature on strategic decision-making. The theoretical foundation for information sources management capability that we discuss in this paper in context to Protein Ontology would help shed light on the information acquisition behavior in the context of technology post-adoption use in the healthcare industry too.

2 Information Sources Management Capability

This study discusses that information sources management is an organizational capability, which, under certain conditions, endows the organization with a technology competitive advantage. Researchers in proteomics domains have generated a substantial body of data literature on protein complexes. The Resource-Based View (RBV) approach [5, 6] primarily concentrates on the impact of organizational resources on the organization's competitive advantage. In their seminal article, which commenced the Dynamic Capabilities (DC) perspective, [7] recognized a wide range of the organization assets and resources, including financial, reputation, structural, technological and institutional. The focus of the RBV approach on organizational resources and capabilities will guide our theoretical thinking for formulating propositions for explaining the process of acquiring information from protein data and information sources using technology resources.

Whereas the RBV approach primarily focused on management of internal capabilities, the DC perspective added the important aspect of external influence on organizational capabilities. According to [7], organizations achieved superior performance if they possessed such organizational capabilities that were flexible in adjusting to environmental dynamics. [5] and [7] emphasized the importance of examining organization capabilities in the context of their historical development. In particular, [5] contended that the organization acquired resources under unique historical circumstances: time and space factors influenced its acquisition of resources. [7] Coined the term 'evolutionary paths' in respect to the processes undergone by the organization's capabilities in space and time. The notion of historical context for development of organizational capabilities agrees with our choice of post-adoption period of technology use. Once a technology is deployed, market or internal conditions dictate necessary adjustments to this technology. While we do not intend to investigate the evolutionary paths followed for collection of protein data by organizations maintaining major protein data sources in this study, we aim to study what approaches we need to employ to optimize integration of major protein data sources. We learn lessons to make technology adjustments through information sources management capability in this section.

In this study, we want to advance the notion of information sources management capability. This capability relies on ownership or access to a combination of technology resources and protein data and knowledge repositories. In addition, information sources management capability encompasses linking diverse data and knowledge repositories that belong to various organizations to have common representation framework. In the next sections we discuss how we applied lessons learnt from information sources management capability to manage and integrate diverse Protein Data and Knowledge Sources.

3 Protein Ontology Conceptual Framework

Advances in technology and the growth of life sciences are generating ever increasing amounts of data. High-throughput techniques are regularly used to capture thousands of data points in an experiment. The results of these experiments normally end up in scientific databases and publications. Although there have been concerted efforts to capture more scientific data in specialist databases, it is generally acknowledged that only 20 per cent of biological knowledge and data is available in a structured format. The remaining 80 per cent of biological information is hidden in the unstructured scientific results and texts. Protein Ontology (PO) provides a common structured vocabulary for this structured and unstructured information and provides researchers a medium to share knowledge in proteomics domain. It consists of concepts, which are data descriptors for proteomics data and the relations among these concepts. Protein Ontology has (1) a hierarchical classification of concepts represented as classes, from general to specific; (2) a list of attributes related to each concept, for each class; and (3) a set of relations between classes to link concepts in ontology in more complicated ways then implied by the hierarchy, to promote reuse of concepts in the ontology. Protein Ontology provides description for protein domains that can be used to describe proteins in any organism. Protein Ontology Framework describes: (1) Protein Sequence and Structure Information, (2) Protein Folding Process, (3) Cellular Functions of Proteins, (4) Molecular Bindings internal and external to Proteins and (5) Constraints affecting the Final Protein Conformation. Protein Ontology uses all relevant protein data sources of information. The structure of PO provides the concepts necessary to describe individual proteins, but does not contain individual protein themselves. A database based on PO acts as instance store for the PO. PO uses data sources include new proteome information resources like PDB, SCOP, and RESID as well as classical sources of information where information is maintained in a knowledge base of scientific text files like OMIM and from various published scientific literature in various journals. PO Database is represented using Web Ontology Language (OWL). PO Database at the moment contains data instances of following protein families: (1) Prion Proteins, (2) B.Subtilis, (3) CLIC and (4) PTEN. More protein data instances will be added as PO is more developed. More details about PO is available at the website: **http://www.proteinontology.info/**

Semantics in protein data is normally not interpreted by annotating systems, since they are not aware of the specific structural, chemical and cellular interactions of protein complexes. Protein Ontology Framework provides specific set of rules to cover these application specific semantics. The rules use only the relationships whose semantics are predefined to establish correspondence among terms in PO. The set of relationships with predefined semantics is: {SubClassOf, PartOf, AttributeOf, InstanceOf, and ValueOf}.

The PO conceptual modeling encourages the use of strictly typed relations with precisely defined semantics. Some of these relationships (like SubClassOf, InstanceOf) are somewhat similar to those in RDF Schema but the set of relationships that have defined semantics in our conceptual PO model is small so as to maintain simplicity of the system. The following is a description of the set of pre-defined semantic relationships in our common PO conceptual model.

SubClassOf: The relationship is used to indicate that one concept is a subclass of another concept, for instance: SourceCell SubClassOf FunctionalDomains. That is any instance of SouceCell class is also instance of FunctionalDomains class. All attributes of FunctionalDomains class (_FuncDomain_Family, _FuncDomain_SuperFamily) are also the attributes of SourceCell class. The relationship SubClassOf is transitive.

AttrributeOf: This relationship indicates that a concept is an attribute of another concept, for instance: _FuncDomain_Family AttributeOf Family. This relationship also referred as PropertyOf, has same semantics as in object-relational databases.

PartOf: This relationship indicates that a concept is a part of another concept, for instance: Chain PartOf ATOMSequence indicates that Chain describing various residue sequences in a protein is a part of definition of ATOMSequence for that protein.

InstanceOf: This relationship indicates that an object is an instance of the class, for instance: ATOMSequenceInstance_10 InstanceOf ATOMSequence indicates that ATOMSequenceInstance_10 is an instance of class ATOMSequence.

ValueOf: This relationship is used to indicate the value of an attribute of an object, for instance: “Homo Sapiens” ValueOf OrganismScientific. The second concept, in turn has an edge, OrganismScientific AttributeOf Molecule, from the object it describes.

4 Comparing GO and PO

Gene Ontology (GO) [8] defines a structured controlled vocabulary in the domain of biological functionality. Characteristics of GO that we believe are most responsible for its success: community involvement; clear goals; limited scope; simple, intuitive structure; continuous evolution; active curation; and early use.

Mining of Scientific Text and Literature is done to generate list of keywords that is used as GO terms. However, querying heterogeneous, independent databases in order to draw these inferences is difficult: The different database projects may use different terms to refer to the same concept and the same terms to refer to different concepts. Furthermore, these terms are typically not formally linked with each other in any way. GO seeks to reveal these underlying biological functionalities by providing a structured controlled vocabulary that can be used to describe gene products, and shared between biological databases. This facilitates querying for gene products that share biologically meaningful attributes, whether from separate databases or within the same database.

Challenges faced while developing GO from unstructured and structured data sources are addressed while developing PO. PO is a conceptual model that aim to support consistent and unambiguous knowledge sharing and that provide a framework for protein data and knowledge integration. PO links concepts to their interpretation, i.e. specifications of their meanings including concept definitions and relationships to other concepts. Apart from semantic relationships defined in Section 3, PO also model relationships like Sequences. By itself semantic relationships described in Section 3, does not impose order among the children of the node. In applications using Protein Sequences, the ability of expressing the order is paramount. Generally Protein Sequences are a collection of chains of sequence of residues, and that is the format Protein Sequences have been represented unit now using various data representations and data mining techniques for bioinformatics. When we are defining sequences for semantic heterogeneity of protein data sources using PO we are not only considering traditional representation of protein sequences but also link Protein Sequences to Protein Structure, by linking chains of residue sequences to atoms defining three-dimensional structure. In this section we will describe how we used a special semantic relationship like *Sequence(s)* in Protein Ontology to describe complex concepts defining Structure, Structural Folds and Domains and Chemical Bonds describing Protein Complexes. PO defines these complex concepts as *Sequences* of simpler generic concepts defined in PO. These simple concepts are *Sequences* of object and data type properties defining them. A typical example of *Sequence* is as follows. PO defines a complex concept of *ATOMSequence* describing three dimensional structure of protein complex as a combination of simple concepts of *Chains*, *Residues*, and *Atoms* as: *ATOMSequence Sequence (Chains Sequence (Residues Sequence (Atoms)))*. Simple concepts defining ATOMSequence are defined as: *Chains Sequence (ChainID, ChainName, ChainProperty)*; *Residues Sequence (ResidueID, ResidueName, ResidueProperty)*; and *Atoms Sequence (AtomID, Atom, ATOMResSeqNum, X, Y, Z, Occupancy, TempratureFactor, Element)*. Thus, PO reflects the structure and relationships of Protein Data Sources.

PO removes the constraints of potential interpretations of terms in various data sources and provides a structured vocabulary that unifies and integrates all data and knowledge sources for proteomics domain. There are seven subclasses of Protein Ontology (PO), called Generic Classes that are used to define complex concepts in other PO Classes: Residues, Chains, Atoms, Family, AtomicBind, Bind, and SiteGroup. Concepts from these generic classes are reused in various other PO Classes for definition of Class Specific Concepts. In PO the notions classification, reasoning, and consistency are applied by defining new concepts or classes from defined generic concepts or classes. The concepts derived from generic concepts are placed precisely into class hierarchy of PO to completely represent information defining a protein complex.

5 Mining Facilitated by Protein Ontology

The Protein Ontology Database is created as an instance store for various protein data using the PO format. PO uses data sources like PDB, SCOP, OMIM and various published scientific literature to gather protein data. PO Database is represented using

XML. PO Database at the moment contains data instances of following protein families: (1) Prion Proteins, (2) B.Subtilis, (3) CLIC and (4) PTEN. More protein data instances will be added as PO is more developed. The PO instance store at moment covers various species of proteins from bacterial and plant proteins to human proteins. Such a generic representation using PO shows the strength of PO format representation.

We used some standard hierarchical and tree mining algorithms [9] on the PO Database. We compared MB3-Miner (MB3), X3-Miner (X3), VTreeMiner (VTM) and PatternMatcher (PM) for mining embedded subtrees and IMB3-Miner (IMB3), FREQT (FT) for mining induced subtrees of PO Data. In these experiments we are mining Prion Proteins dataset described using Protein Ontology Framework, represented in OWL. For this dataset we map the OWL tags to integer indexes. The maximum height is 1. In this case all candidate subtrees generated by all algorithms would be induced subtrees. **Figure 1** shows the time performance of different algorithms. Our original MB3 has the best time performance for this data.

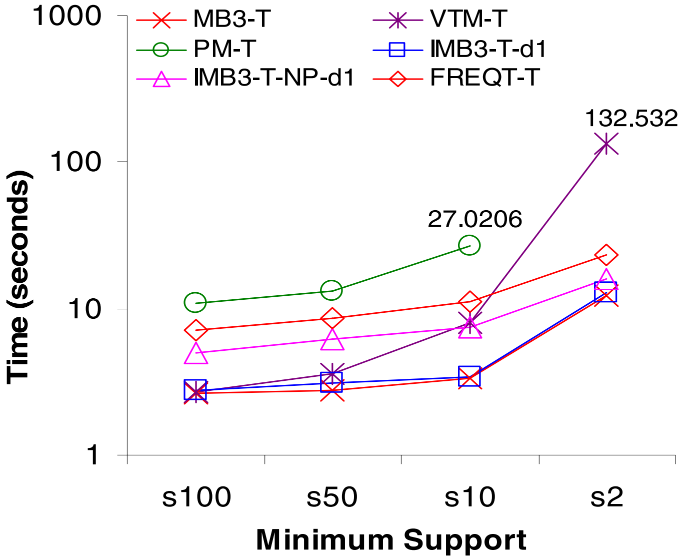


Fig. 1. Time Performance for PO Data

Quite interestingly, the subtrees generated of the PO dataset represented in OWL are same for every algorithm. Therefore the conceptual framework of PO provides a powerful hierarchical classification of concepts, which provides consistency and accuracy in observations of various analysis and reasoning methodologies.

6 Conclusion

Protein Ontology (PO) provides a unified vocabulary for capturing declarative knowledge about protein domain and to classify that knowledge to allow reasoning.

Information captured by PO is classified in a rich hierarchy of concepts and their inter-relationships. PO is compositional and dynamic, relying on notions of classification, reasoning, consistency, retrieval and querying. In PO the notions classification, reasoning, and consistency are applied by defining new concepts or classes from defined generic concepts or classes. The concepts derived from generic concepts are placed precisely into class hierarchy of Protein Ontology to completely represent information defining a protein complex. Protein Ontology (PO) is the first ever work to integrate protein data based on data semantics describing various phases of protein structure. PO Database at the moment contains data instances of following protein families: (1) Prion Proteins, (2) B.Subtilis, (3) CLIC and (4) PTEN. More protein data instances will be added as PO is more developed. The PO instance store at moment covers various species of proteins from bacterial and plant proteins to human proteins. Such a generic representation using PO shows the strength of PO format representation.

References

[1] Sidhu, A. S., T. S. Dillon, et al. (2006). Ontology for Data Integration in Protein Informatics. In: Database Modeling in Biology: Practices and Challenges. Z. Ma and J. Y. Chen. New York, NY, Springer Science, Inc.: In Press.

[2] Sidhu, A. S., T. S. Dillon, et al. (2006). Protein Ontology Project: 2006 Updates (Invited Paper). Data Mining and Information Engineering 2006. A. Zanasi, C. A. Brebbia and N. F. F. Ebecken. Prague, Czech Republic, WIT Press.

[3] Sidhu, A. S., T. S. Dillon, et al. (2005). Ontological Foundation for Protein Data Models. First IFIP WG 2.12 & WG 12.4 International Workshop on Web Semantics (SWWS 2005). In conjunction with On The Move Federated Conferences (OTM 2005). Agia Napa, Cyprus, Springer-Verlag. Lecture Notes in Computer Science (LNCS).

[4] Sidhu, A. S., T. S. Dillon, et al. (2005). Protein Ontology: Vocabulary for Protein Data. 3rd IEEE International Conference on Information Technology and Applications (IEEE ICITA 2005). Sydney, IEEE CS Press. Volume 1: 465-469.

[5] Barney, J.B. (1991), "Organization Resources and Sustained Competitive Advantage," Journal of Management, 17, 1, 99-120.

[6] Peteraf, M. A. and Barney, J. B. (2003). “Unraveling the Resource-Based Tangle,” Managerial and Decision Economics, 24, 309-323.

[7] Teece, D.J., Pisano, G., and Shuen, A. (1997). "Dynamic Capabilities and Strategic Management," Strategic Management Journal, 18, 7, 509-533.

[8] GO Consortium and S. E. Lewis (2004). "Gene Ontology: looking backwards and forwards." Genome Biology 6(1): 103.1-103.4.

[9] Tan, H., T.S. Dillon, et. al. (2006). IMB3-Miner: Mining Induced/Embedded Subtrees by Constraining the Level of Embedding. Accepted for Proceedings of PAKDD 2006.

Semantic Web Modeling for Virtual Organization: A Case Study in Logistics*

Liao Lejian and Zhu Liehuang

School of Computer Sciences and Engineering
Beijing Institute of Technology, Postcode 100081, Beijing, China
liaolj@bit.edu.cn, liehuangz@bit.edu.cn

Abstract. Cross-organizational interoperability and coordination are major challenges to Virtual Organization(VO) applications. In this paper, a semantic Web enabled multi-agent platform for logistic VO supporting is proposed. The issue of extending OWL with multi-attribute constraints for VO modeling is addressed. A constraint rule language *SWOCRL* is proposed which is based on OWL and SWRL with constraint extension and class-scoped restriction. Important VO concepts such as organizations, activities, resources and their logistic specializations are described with OWL plus *SWOCRL.*

Keywords: semantic Web, agent, virtual organization modeling, constraint rule, ontology.

1 Introduction

Web-based computer platforms that supports VO faces challenges of finding suitable business partners from Internet, and having heterogeneous agents to interact and coordinate with each other. Semantic Web (SW) combined with multi-agent systems is a promising technology for solving these challenging problems. For SW to solve these problems, it is necessary for SW-enabled VO agents to understand common conceptualization across business domains. VO conceptualization extensively involves modeling of such concepts as organizations, time and space, processes and activities, physical resources, interaction and negotiation, policies and agreements. Such conceptualization typically involves representation of complex constraints such as multi-attribute constraints of the concepts. In SW pyramid, information modeling mainly involves RDF(S) layer, ontology layer, and logic layer. The modeling of general multi-attribute constraints is not supported in current ontology layer language OWL[1]. Syntactically SW rule language RuleML[2] should cover any datalog rules including constraint expressions. But semantically constraint expressions are interpreted according to built-in domain theories which are beyond the expressibility of datalog rules. In this paper, we propose a constraint language which is based on OWL and SWRL[3], and demonstrate its applicability in VO modeling with our urgent logistic scenario.

* This work is funded by National Science Foundation of China under grant No. 60373057.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 602 – 608, 2006.

© Springer-Verlag Berlin Heidelberg 2006

2 Scenario and System

The following is a scenario for our investigation. Suppose that a large-area devastating disaster, like the Pacific tsunami occurred in Asia in 2004, erupted in some remote area of China. A government committee is set up to organize all the urgent rescuing, recovery, and aiding activities in the disaster area. Subordinate to the committee, a group G-Logistics is responsible for the supply of the necessity goods and equipments to the disaster area. The organizational coordination activities around the disaster-rescuing include: 1) G-Logistics searches for merchants, manufacturers and logistic companies for the demanded goods and their shipment, and negotiates with the selected candidates. 2) Merchants and Manufacturers publish their products and services, and negotiate with customers like G-Logistics for sales. 3) Logistic companies publish their businesses and negotiate with interested customers such as G-Logistics for shipment tasks. To perform a shipment task, a logistic company may negotiate with warehouses for intermediate storage in the process of transportation, and with the gas-stations along the highway for intermediate refueling. 4) Warehouses provide relayed storage for the goods during the transportation process. They periodically publish their spare spaces for goods storage. 5) Gas-stations periodically publish their provision quota in a day, a week, or a month, with policies for overly large mount of consumption request.

With the urgent logistics background, we design a service-oriented agent-based VO supporting platform. The architecture consists of a set of service agents and task agents. A task agent, such as the organizational agent for G-Logistics in the scenario, is based on BDI model which has beliefs, goals, actions, and strategies in its body. A service agent, such as those for merchants, logistic company, gas-stations in the scenario, publish their service as semantic Web service descriptions with OWL-S and behavior pattern as constraints. Some service agents may at the same time be task agents in that they act as clients to other service agents during service execution. As a kernel component of the VO supporting platform, an OWL-S enhanced semantic UDDI database stores all the service items for service discovery. Service discovery is invoked with a semantic service request issued by a task agent to a service matchmaker that matches the request against service items in the UDDI database. In our VO supporting platform, several features are added to the original semantic Web service matchmaking algorithms. Other components of the VO supporting platform include Web-service wrapped agent communication infrastructure, semantic Web and ontology management, editing and reasoning modules, general VO ontology, agent registration and management modules, knowledge base tools(currently f-Logic and Fuzzy Clips), semantic-web service enabled model base for authorized agents, GIS sever, security management modules, and etc.

3 *SWOCRL*: A Language for SW Modeling with Object Constraints

A starting point of this work is to extend rigid description-logic formulas of OWL with more flexible object constraints that are common in VO modeling. We propose a constraint languages *SWOCRL*(Semantic Web Object Constraint Rule Language) which is based on OWL and SWRL. It relies on the conceptual structure defined by OWL assertions, and uses rules to infer object structures and to impose constraints on object attributes. *SWOCRL* extends SWRL by allowing

1) Some attributes (unique roles) and attribute paths as constraint variables. An RDF statement *rdfs:subClassOf* (*swocrl: ConstraintAttribute*, *owl:UniqueProperty*) defines a special class *swocrl: ConstraintAttribute* of such constraint attributes.

2) Multi-attribute constraints as atoms in both antecedents and consequents of rules. Variables introduced in antecedents are taken to be a universal variable; Variables introduced in consequents are taken to be existential ones.

SWOCRL then adds the following syntax rules to SWRL:

atom ::= *constraint*
constraint::=*predicate-name* '(' {*d-object*} ')'
d-object ::= *d-path-exp*
d-path-exp ::= *d-attribute* | *d-attribute* '.' *d-path-exp*
i-object :: =*i-path-exp*
i-path-exp ::= *i-attribute* | *i-attribute* '. ' *i-path-exp*

SWOCRL specializes SWRL such that a *SWOCRL* rule only specifies assertions of just one class, featuring it as an object-centered constraint language. Such specialization is desirable to circumscribe the complexity of rule reasoning.The specialization includes the following restrictions to the above syntax:

1) The first atom in the antecedent must be fixed as *class-name* '(' *i-variable* ')' which indicates the class that the rule asserts about.
2) For the following atoms, a unary class description atom must have an instantiated argument, i.e., either a constant individual, or an *i-variable* that occurs in a preceding atom; a binary property atom must have the first argument instantiated.

A *SWOCRL* rule in such abstract syntax can be written as first-order rule as follows:

$$\forall X_1,\ldots,X_m\ c(X_1) \wedge p_1(.)\wedge\ldots\wedge p_k(.) \rightarrow \exists Y_1,\ldots,Y_n\ q_1(.)\wedge\ldots\wedge q_l(.)$$

Where c is a class name, $p_1,\ldots, p_k$, $q_1,\ldots, q_l$ are either class name, or property name, or constraint relation. For clarity, this first-order rule form is used in the following section. Universal(Existential) variables are identified as X(Y)-prefixed names.

4 VO Ontological Modeling

Here we show some DL-form concept descriptions and associated *SWOCRL* constraint rules that are typical in VO modeling in general and urgent logistics in particular. Simplifications are made for clarity and space limitation reasons.

4.1 Organizational Modeling

We view an organization as a service actor with certain composition structure and functioning for some services. The following lists DL axioms for organization, logistic-organization and warehouse as well as their functional services, starting from the concept of *Service-actor.*

$\textit{Service-actor} \subseteq \forall \textit{has-service. Actor-service}$

$\textit{Actor-service} \subseteq \forall \textit{has-action. Action} \cap \forall \textit{has-goal.Goal}$

$\textit{Organization} \subseteq \textit{Service-actor} \cap \forall \textit{has-service.Organization-service} \cap$
$\forall \textit{has-subordinate.Organization} \cap \forall \textit{subordinate-to.Organization} \cap$
$\forall \textit{has-position.Organizational-position} \cap \forall \textit{located-in. Location}$

Note that *subordinate-to* is the inverse property of *has-subordinate.*

$\textit{Organization-service} \subseteq \textit{Actor-service} \cap \forall \textit{has-action.Organization-action} \cap$
$\forall \textit{has-goal.Organization-goal}$

The following is logistic specialization of general organization conception.

$\textit{Logistic-organization} \subseteq \textit{Organization} \cap \forall \textit{has-service.Logistic-service}$

$\textit{Logistic-service} \subseteq \textit{Organization-service}$

$\textit{Warehouse} \subseteq \textit{Logistic-organization} \cap \forall \textit{has-service. Warehouse-service}$

$\textit{Warehouse-service} \subseteq \textit{Logistic-service} \cap \forall \textit{stored-goods.Goods} \cap$
$=1 \textit{current-amount.Integer} \cap \forall \textit{recent-storage-plan. Warehouse-plan} \cap$
$=1 \textit{has-storage-capacity.Integer}$

$\textit{Warehouse-plan} \subseteq =1 \textit{ time-period Time} \cap =1 \textit{input-amount. Integer} \cap$
$=1 \textit{output-amount. Integer}$

Roles *input-amount* and *output-amount* are defined as constraint attributes:
input-amount: *ConstraintAttribute, output-amount*: *ConstraintAttribute,*

A capacity constraint for the storage in a time is imposed on a warehouse, i.e. Current amount + input - output < storage capacity. Such constraint is expressed in the following *SWOCRL* rule with built-in constraints $x+y=z$ and $x \geq y$.

$\forall \textit{X-ws,X-ws,X-sc,X-sp,X-ca}$
$\textit{Warehouse-service}(\textit{X-ws}) \wedge \textit{has-storage-capacity}(\textit{X-ws,X-sc}) \wedge$
$\textit{has-storage-plan}(\textit{X-ws, X-sp}) \wedge \textit{current-amount}(\textit{X-ws, X-ca})$
$\rightarrow \exists \textit{Y1,Y2}$
$\textit{Y1} = \textit{X-ca} + \textit{X-sp.input-amount} \wedge \textit{Y2} = \textit{X-sp.output-amount} + \textit{X-sc} \wedge \textit{Y1} \geq \textit{Y2}$

4.2 Activity Modeling

In our work, activities are modeled according to their goals, participants, compositional and temporal relation.

$\textit{Activity} \subseteq \forall \textit{has-service.Actor-service} \cap \forall \textit{has-role.Actor} \cap$
$\forall \textit{has-subactivity.Activity} \cap \forall \textit{succeed-to. Activity}$

Activities can further have time attached for refined temporal description.

$\textit{Timed-activity} \subseteq \textit{Activity} \cap \forall \textit{in-period.Time-Interval}$

$\textit{Time-Interval} \subseteq =1 \textit{start-time.Time-point} \cap =1 \textit{ end-time.Time-point} \cap$
$=1 \textit{duration. Time-Duration}$

start-time:*ConstraintAttribute,end-time*:*ConstraintAttribute,*
duration:*ConstraintAttribute*

The relation of the start time, end time, and duration of a time interval can be represented as *SWOCRL* rule:

$\forall \textit{X-ti Time-interval } (\textit{X-ti}) \rightarrow \textit{X-ti.start-time} + \textit{X-ti.duration} = \textit{X-ti. end-time}$

Specific to the logistic field, we have activities such as movements and transportation.

$\textit{Movement} \subseteq \textit{Timed-activity} \cap \forall\textit{along- path.Path} \cap {=}\textit{1average-velocity.Velocity}$

$\textit{Path} \subseteq \textit{Spatial-entity} \cap \forall\textit{on-path.Location} \cap {=}1\textit{start-location.Location} \cap$
$\quad {=}1\textit{end-location. Location} \cap {=}1\ \textit{length.Length-metric}$

start-location and *end-location* are sub-properties of *on-path:*

$\textit{start-location} \subseteq \textit{on-path},\ \textit{end-location} \subseteq \textit{on-path}$

The following SWORCL rules express temporal-spatial assertions for *Movement* that the actor is at the starting location at the start, and at the end location in the end.

$\forall\textit{X-mv,X-rl,X-ap,X-sl,X-el,X-tp}$
$\quad \textit{Movement}(\textit{X-mv}) \wedge \textit{has-role}(\textit{X-mv, X-rl}) \wedge \textit{along-path}(\textit{X-mv, X-ap}) \wedge$
$\quad \textit{start-location}(\textit{X-ap, X-sl}) \wedge \textit{end-location}(\textit{X-ap, X-el}) \wedge \textit{in-period}\ (\textit{X-mv, X-tp})$
$\quad \rightarrow \textit{at-location}(\textit{X-rl, X-sl, X-tp.start-time}) \wedge \textit{at-location}(\textit{X-rl, X-el, X-tp.end-time})$

For *Movement* there would be another rule to specify the relation between the length, time and velocity, but omitted here.

$\forall\textit{X-mv, X-ap, X-lt, X-av, X-tp}$
$\quad \textit{Movement}(\textit{X-mv}) \wedge \textit{along-path}(\textit{X-mv, X-ap}) \wedge \textit{length}(\textit{X-ap, X-lt}) \wedge$
$\quad \textit{average-velocity(X-mv,X-av)} \wedge \textit{in-period}\ (\textit{X-mv, X-tp})$
$\quad \rightarrow \textit{times-metric}(\textit{X-lt}, \textit{X-av}, \textit{X-tp.duration})$

Where *times-metric*(*X-lt*, *X-av*, *X-tp.duration*) denotes the metric counterpart of constraint $\textit{X-lt} \times \textit{X-av} = \textit{X-tp.duration}$. The following are axioms about logistic transportation.

$\textit{Transportation} \subseteq \textit{Movement} \cap \forall\textit{has-actor. Transportation-actor} \cap$
$\quad \forall\textit{with-traffic.Traffic-System} \cap \forall\textit{has-load. Transportation-load}$

$\textit{Traffic-system} \subseteq \textit{Facility} \cap\forall\textit{with-traffic-vehicle.Vehicle} \cap \forall\textit{in-traffic-line.Traffic-line}$

$\textit{HY-Transportation} \subseteq \textit{Transportation} \cap\forall\textit{with-traffic. Highway}$

$\textit{Highway} \subseteq \textit{Traffic-system} \cap\forall\textit{with-traffic-vehicle.Automobile} \cap$
$\quad \forall\textit{in-traffic-line.Road-line}$

The following rule denote a constraint that for highway transportation the *Path* of the *HY-Transportation* must be in line with the *Road-line*:

$\textit{X-ht, X-ap, X-hy, X-rl HY-Transportation}\ (\textit{X-ht}) \wedge \textit{along-path}(\textit{X-ht, X-ap}) \wedge$
$\qquad \textit{with-traffic}(\textit{X-ht, X-hy}) \wedge \textit{in-traffic-line}(\textit{X-hy, X-rl})$
$\quad \rightarrow \textit{in-line-with}(\textit{X-ap, X-rl})$

Where *in-line-with* is a relation between two instances that is interpreted by another rule, which is omitted here.

4.3 Resource Modeling

The concept of *resources* is in widespread uses in VO trading activities as well as in grid computing [4,5]. By resources we mean objects in contexts of activities that use them. An especially important concept in VO is a quantity volume of resources in which the quantity rather than the individual elements of a set of entities is concerned, such as 20 trucks or 2 millions of quilts. We view a quantified resource as a first-order individual that affiliates the elements of the collection through a multi-value property, named *has-element.*

$$\textit{Resource} \subseteq \forall \textit{has-element. Element-Class} \cap \forall \textit{has-context Resource-Context}$$

Where *Element-Class* is supposed to be the top class of resource elements of concern; *has-context* specifies the context features related to the activity using the resource. The features are modeled as subclasses of *Resource-Context* which contains attributes on which such features depend, as described in below. Several main features of this sort are *allocability*, *sharability*, *reusability*, and *dividability*. *Allocability* indicates if a resource can be allocated independently to an activity. For example, a classroom can be allocated to a class, but a number of classroom seats cannot. *Sharability* indicates if a resource can be allocated to more than one activities in the same time. A highway is sharable to many vehicles with respect to transportation activities, while a classroom can only be used by one class at any time. *Reusability* indicates if a resource can be reused to other activities after its use by one activity. A classroom is reusable w.r.t to class while a gallon of fuel will consume away after it is used out. *Dividability* indicates if a resource can have its parts allocated to other activities. A gallon of fuel can be divided and allocated to two vehicles for running, while a vehicle cannot be divided into two pieces while still perform transportation respectively. These features are modeled as subclasses of *Resource-Context.*

$$\begin{aligned}
&\textit{Resource-Context} \subseteq {=}1\textit{with-activity.Activity}\\
&\textit{Resource-Context} \equiv \textit{Allocable-R-Ctx} \cup \textit{Non-allocable-R-Ctx}\\
&\textit{Allocable-R-Ctx} \cap \textit{Non-allocable-R-Ctx} = \Phi\\
&\textit{Resource-Context} \equiv \textit{Sharable-R-Ctx} \cup \textit{Non-sharable-R-Ctx}\\
&\textit{Sharable-R-Ctx} \cap \textit{Non-sharable-R-Ctx} = \Phi\\
&\textit{Resource-Context} \equiv \textit{Reusable-R-Ctx} \cup \textit{Non-reusable-R-Ctx}\\
&\textit{Reusable-R-Ctx} \cap \textit{Non-reusable-R-Ctx} = \Phi\\
&\textit{Resource-Context} \equiv \textit{Dividable-R-Ctx} \cup \textit{Non-dividable-R-Ctx}\\
&\textit{Dividable-R-Ctx} \cap \textit{Non-dividable-R-Ctx} = \Phi
\end{aligned}$$

The definition of resource contexts as a multi-value property indicates the fact that the activity-related resource features are not uniform for one resource type. For different activities a resource may exhibit opposite features. For example, a bus is sharable to different passengers but not sharable between different running routes. This can be represented as follows:

$$\begin{aligned}
&\textit{Bus-Resource} \subseteq \textit{Resource} \cap {=}1\textit{has-element.Bus}\\
&\textit{Bus-resource} \subseteq \forall\textit{has-context.}(\neg\forall\textit{with-activity.Passenger-Riding} \cup \textit{Sharable-R-Ctx})\\
&\textit{Bus-resource} \subseteq \forall\textit{has-context.}(\ \neg\forall\textit{with-activity.Route-allocation} \cup \textit{Non-sharable-R-Ct}\mathrm{x})
\end{aligned}$$

5 Conclusions

The realization of VO requires computational supports of different distributed Internet computing such as agents and grid[6]. Aiming at VO applications, we propose a SW modeling language *SWOCRL* which allows the representation of object centered constraint rules. It is based on OWL and SWRL and specializes SWRL rules to the scope of single-class specification thus avoid combinatorial complexity of multi-class instances and gains reasoning scalability, while extends it with multi-attribute constraints for modeling expressibility. In [7], the importance of expressive constraints for SW was realized and a constraint language was defined with SW ontology. By comparison *SWOCRL* is motivated as a constraint extension of OWL for VO modeling. Some interesting VO concepts such as organizations, activities, contracts, interactions and their logistic specializations are described here with OWL plus *SWOCRL.* Ontological modeling of enterprise activities was systematically done in PSL[8]. But PSL is written in KIF rules rather than description logics, which makes it not interoperable with current semantic web formulation at conceptual level since the conceptual structure of the ontology is not explicit. For example, they cannot be feasibly reused by the existing semantic web service matchmaking algorithm[9].

Further work will investigate the issue of the modeling of complex interaction and interaction protocol reuse.

References

1. P. F. Patel-Schneider, P. Hayes, and I. Horrocks, OWL web ontology language semantics and abstract syntax. Recommendation 10 February 2004, W3C, 2004
2. RuleML. Rule markup language initiative, 2004. http://www.ruleml.org, 2004
3. I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission, May 2004. http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/, 2004
4. H. Zhuge, The Knowledge Grid, World Scientific Publishing Co., Singapore, 2004
5. H. Zhuge, Resource Space Grid: Model, Method and Platform, Concurrency and Computation: Practice and Experience, 16 (14) (2004) 1385-1413.
6. Ian T. Foster: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. CCGRID 2001: 6-7
7. P. Gray, K. Hui, A. Preece. Mobile Constraints for Semantic Web Applications. In M Musen, B Neumann, & R Studer (eds) Intelligent Information Processing, Kluwer, pages 117-128, 2002.
8. M. Gruninger. PSL 2.0 Ontology – Current Theories and Extensions. http://www.nist.gov/psl/psl-ontology/, 2003
9. M. Paolucci, et al.: Semantic Matching of Web Services Capabilities. Proceedings of the 1st International Semantic Web Conference (ISWC 2002), Sardinia (Italy), Lecture Notes in Computer Science, Vol. 2342. Springer Verlag (2002)

A PSO-Based Web Document Query Optimization Algorithm

Ziqiang Wang, Xin Li, Dexian Zhang, and Feng Wu

School of Information Science and Engineering,
Henan University of Technology, Zheng Zhou 450052, China
wzqagent@xinhuanet.com

Abstract. The particle swarm optimization(PSO) algorithm is a robust stochastic evolutionary algorithm based on the movement and intelligence of swarms.To efficiently retrieve relevant documents from the explosive growth of the Internet and other sources of information access,a PSO-based algorithm for Web document query optimization is presented. Experimental results show that the proposed algorithm can improve the precision of document retrieval markedly compared with relevant feedback and genetic algorithm.

1 Introduction

With the rapid development of Internet, information on the Internet is increasing exponentially.As a consequence, the role of information retrieval (IR) systems is becoming more important.One of the most important and difficult operations in information retrieval is to generate queries that can succinctly identify relevant documents and reject irrelevant documents. In order to get good retrieval performance,there has been a growing interest in applying genetic algorithm(GA) to the information retrieval domain with the purpose of optimizing document descriptions and improving query formulation[1,2].The main advantages of GA lie in its global convergence,inherent parallel search nature,and great robustness.However,owing to the slow convergence for each generation,a revised evolutionary algorithm to improve the convergence efficiency is needed for superior Web document query optimization.

Recently, Eberhart and Kennedy suggested a particle swarm optimization (PSO) based on the analogy of swarm of bird[3]. The main advantages of the PSO algorithm are summarized as: simple concept, easy implementation, robustness to control parameters, and computational efficiency when compared with mathematical algorithm and other heuristic optimization techniques. The original PSO has been applied to a learning problem of neural networks and function optimization problems, and efficiency of the method has been confirmed. In this paper, the objective is to investigate the capability of the PSO algorithm for Web document query optimization in the context of information retrieval.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 609–615, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Particle Swarm Optimization (PSO) Algorithm

The PSO is a population based optimization technique[3], where the population is called a swarm. A simple explanation of the PSO's operation is as follows. Each particle represents a possible solution to the optimization task.During each iteration each particle accelerates in the direction of its own personal best solution found so far, as well as in the direction of the global best position discovered so far by any of the particles in the swarm. This means that if a particle discovers a promising new solution, all the other particles will move closer to it, exploring the region more thoroughly in the process.

Let n denotes the swarm size.Each individual particle $i(1 \leq i \leq n)$ has the following properties: a current position x_i in search space, a current velocity v_i, and a personal best position p_i in the search space, and the global best position p_{gb} among all the p_i.During each iteration, each particle in the swarm is updated using the following equation .

$$v_i(t+1) = k[w_i v_i(t) + c_1 r_1(p_i - x_i(t)) + c_2 r_2(p_{gb} - x_i(t))] \tag{1}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

where c_1 and c_2 denote the acceleration coefficients, and r_1 and r_2 are random numbers uniformly distributed within [0,1].

The value of each dimension of every velocity vector v_i can be clamped to the range $[-v_{max}, v_{max}]$ to reduce the likelihood of particles leaving the search space. The value of v_{max} chosen to be $k \times x_{max}$(where $0.1 \leq k \leq 1$).Note that this does not restrict the values of x_i to the range $[-v_{max}, v_{max}]$.Rather than that, it merely limits the maximum distance that a particle will move.

Acceleration coefficients c_1 and c_2 control how far a particle will move in a single iteration. Typically, these are both set to a value of 2.0, although assigning different values to c_1 and c_2 sometimes leads to improved performance.The inertia weight w in Equation (6) is also used to control the convergence behavior of the PSO.Typical implementations of the PSO adapt the value of w linearly decreasing it from 1.0 to near 0 over the execution. In general, the inertia weight w is set according to the following equation[4]:

$$w_i = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \cdot iter \tag{3}$$

where $iter_{max}$ is the maximum number of iterations, and $iter$ is the current number of iterations.

In order to guarantee the convergence of the PSO algorithm, the constriction factor k is defined as follows:

$$k = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \tag{4}$$

where $\varphi = c_1 + c_2$ and $\varphi > 4$.

The PSO algorithm performs the update operations in terms of Equation (1) and (2) repeatedly until a specified number of iterations have been exceeded, or velocity updates are close to zero.The quality of particles is measured using a fitness function which reflects the optimality of a particular solution.

3 The PSO Algorithm for Web Document Query

The proposed system is based on a vector space model[5] in which both documents and queries are represented as vectors. The goal of the PSO algorithm is to find an optimal set of documents which best match the user's need by exploring different regions of the document space simultaneously.The detail steps of the PSO-based Web document query algorithm are described as follows.

3.1 The Definition and Encoding of the Query Particle

The first step toward implementation of the PSO is the definition the particle of swarm(population) to be evolved (i.e.,solution space).In this study,the particle is represented by query vector space. Each query particle representing a query is of the form:

$$Q_u = (q_{u1}, q_{u2}, \cdots, q_{uT}) \tag{5}$$

where T is total number of stemmed terms automatically extracted from the documents, q_{ui} is the weight of the ith term in Q_u and is represented by a real value and defines the importance of the term in the considered query. Initially, a term weight q_{ui} is computed as the following formula[6]:

$$q_{ui} = \frac{(1 + log(tf_{ui})) \cdot log(\frac{N}{n_i})}{\sqrt{\sum_{k=1}^{T}((1 + log(tf_{ui})) \cdot log(\frac{N}{n_i}))^2}} \tag{6}$$

where tf_{ui} is the frequency of term t_i in document d_u, N is the total number of documents, and n_i is the number of documents containing the term t_i.

Therefore, particle $i's$ position at iteration 0 can be represented as the vector $Q_i^0 = (q_{i1}^0, \ldots, q_{iT}^0)$ where T is total number of stemmed terms automatically extracted from the documents.The velocity of particle i(i.e.,$V_i^0 = (v_{i1}^0, \ldots, v_{iT}^0)$) corresponds to the term weight update quantity,the velocity of each particle is created at random.The elements of position and velocity have the same dimension.

3.2 Fitness Function

A fitness is assigned to each query in the population. This fitness represents the effectiveness of a query during the retrieving stage. Its definition is as follows:

$$F(Q_u^{(s)}) = \frac{1}{N} \cdot \frac{\sum_{d_j \in D_r^{(s)}} Sim(d_j, Q_u^{(s)})}{\sum_{d_j \in D_{nr}^{(s)}} Sim(d_j, Q_u^{(s)})} \tag{7}$$

where N is the total number of documents, $D_r^{(s)}$ is the set of relevant documents retrieved at the generation(s) of the PSO,d_j is the jth document,$D_{nr}^{(s)}$ is the set of non-relevant documents retrieved at the generation(s) of the PSO, and $Sim(d_j, Q_u^{(s)})$ is a similar measure function defined as follows:

$$Sim(d_j, Q_u^{(s)}) = Cos(d_j, Q_u^{(s)}) = \frac{\sum_{i=1}^{T}(q_{ui}^{(s)} \cdot d_{ji})}{\sqrt{\sum_{i=1}^{T} q_{ui}^2} \cdot \sqrt{\sum_{i=1}^{T} d_{ji}^2}} \tag{8}$$

3.3 Personal and Global Best Position Computation

Each particle i memorizes its own $F(Q_i^{(s)})$ value and chooses the maximum one, which has been better so far as personal best position $P_i^{(s)}$ where s denotes the iteration number.The particle with the best F value among $P_i^{(s)}$ is denoted as global best position $P_{gb}^{(s)}$.Note that in the first iteration,each particle i is set directly to $P_i^{(0)}$, and the particle with the best F value among $P_i^{(0)}$ is set to $P_{gb}^{(0)}$. Since each particle initial position is the only location encountered by each particle at the run's start,this position becomes each particle's respective personal best position $P_i^{(0)}$.The first global best position $P_{gb}^{(0)}$ is then selected from among these initial positions.

3.4 Update the Position and Velocity of Each Particle

Calculate the fitness value of each particle in the population using the fitness function F given by Equation(7).Compare each particle's fitness value with its personal best position $P_i^{(s)} = (p_{i1}^{(s)}, \ldots, p_{iT}^{(s)})$,the global best position is denoted as $P_{gb}^{(s)}$.Modify the member velocity V_i of each particle i according to the following formulation:

$$V_i^{(s+1)} = k[w_i V_i^{(s)} + c_1 r_1 (P_i^{(s)} - V_i^{(s)}) + c_2 r_2 (P_{gb}^{(s)} - V_i^{(s)})] \tag{9}$$

If $V_i^{(s+1)} > V_{max}$, then $V_i^{(s+1)} = V_{max}$.

Based on the updated velocities, each individual(particle) changes its position according to he following formulation:

$$Q_i^{(s+1)} = Q_i^{(s)} + V_i^{(s+1)} \tag{10}$$

The personal best position $P_i^{(s)}$ of individual at iteration $(s+1)$ is updated as follows:

If $F(Q_i^{(s+1)}) > F(Q_i^{(s)})$ then $P_i^{(s+1)} = Q_i^{(s+1)}$;

If $F(Q_i^{(s+1)}) < F(Q_i^{(s)})$ then $P_i^{(s+1)} = Q_i^{(s)}$;

where $F(Q_i^{(s)})$ denotes the fitness function evaluated at the iteration number s. Meanwhile,the global best position P_{gb} at iteration $(s+1)$ is set as the best evaluated position among $P_i^{(s+1)}$.

3.5 Local Search Procedure

To reinforce the local search abilities of PSO, our algorithm adopts a neighborhood-based local search procedure to find a better query vector near the original query vector after applying the PSO algorithm.Let $Q_u^{(s)+}$ and $Q_u^{(s)-}$ be the neighbors of the query vector $Q_u^{(s)}$, their definition is as follows:

$$q_{ui}^{(s)+} = q_{ui}^{(s)} \cdot (1+\beta) \tag{11}$$

$$q_{ui}^{(s)-} = q_{ui}^{(s)} \cdot (1 - \beta) \tag{12}$$

where the value of β decides the ratio of increase or decrease. Each weight in a query vector generates two neighboring vectors. From all neighboring vectors, the vector $Q_u^{(s)}(new)$ which has the best fitness function value is selected.If $avg(Q_u^{(s)}(new))$ is larger than $Q_u^{(s)}$,then the $Q_u^{(s)}$ is replaced by $Q_u^{(s)}(new)$.

3.6 Retrieved Relevant Documents Merging

At each generation of PSO, these retrieved relevant documents by all the individual queries of the query population are merged to a single document list, and presented to user. Our adopted merging methods according to following range formula:

$$Rel^{(s)}(d_j) = \sum_{Q_u^{(s)} \in Pop^{(s)}} F(Q_u^{(s)}) \cdot RSV(Q_u^{(s)}, d_j) \tag{13}$$

where $Pop^{(s)}$ is the population at the generation(s) of the PSO, $RSV(Q_u^{(s)}, d_j)$ is the retrieval status value(RSV) of the document d_j for the query $Q_u^{(s)}$ at the generation(s) of the PSO.

3.7 Stopping Criteria

The PSO algorithm is terminated if the best evaluation value P_{gb} is not obviously improved or the iteration number s approaches to the predefined maximum iteration.

4 Experimental Results and Comparison

To test the performance of the proposed PSO query optimization algorithm, our experiment used the best known the TREC collections, namely TREC D1&D2[7],which contain about 742,600 documents, One of the principal reasons for the choice of these collections was that they had been used elsewhere for query optimization experiments. We retrieved the top-ranked 1000 documents for 50 queries, and evaluated the results of the retrieval via the classical measures of recall and precision.

The parameter settings of the PSO algorithm are as follows: the ratio β of increase(decrease) in local search is set 0.05 , and the number of iterations is fixed at 5.The comparison of our PSO algorithm with classical relevant feedback approach[8] and genetic algorithm(GA)[2] is shown in Figure 1. From results of Figure 1,we can see that our PSO query optimization algorithm can improve the precision of document retrieval markedly compared with relevant feedback and genetic algorithm. The reason is that we designed corresponding position and velocity of each particle updating operation according to itself characteristics of information retrieval, and used local search method to speed up finding a

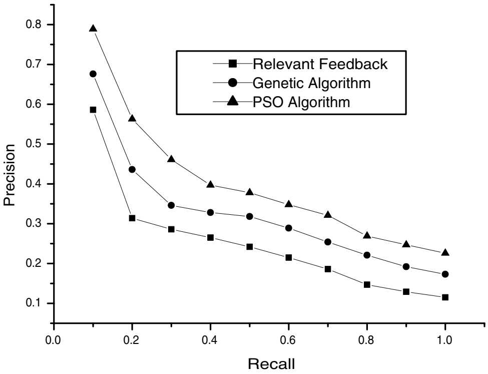


Fig. 1. Performance comparison of three optimization algorithms

Table 1. Comparison the Number of Relevant Document

Algorithm	Iter-1	Iter-2	Iter-3	Iter-4	Iter-5
PSO	110(110)	68(178)	52(230)	57(287)	45(332)
GA	96(96)	64(160)	55(215)	51(266)	32(298)

better query vector near the original query vector after applying the PSO operation. Therefore, our PSO query optimization algorithm markedly improved the precision of document retrieval.

In addition, we also compare the number of relevant document retrieved using PSO and GA. Table 1 gives the number of relevant document retrieved at each iteration of the PSO and GA, and the cumulative total number at that point. We can clearly see that our PSO more effective than GA in retrieving relevant documents. Indeed the cumulative total number of relevant documents using PSO through all the iterations is higher than using GA. Therefore, our proposed PSO query optimization algorithm efficiently improves the performance of the query search.

5 Conclusions and Future Works

In this paper, a PSO-based algorithm for Web document query optimization is presented. Experimental results show that the proposed algorithm can improve the precision of document retrieval markedly compared with relevant feedback and genetic algorithm. In future, we plan to combine other efficient heuristics methods to further improve the document retrieval performance.

Acknowledgement

This work was supported by the Natural Science Foundation of Henan Province under Grant No.0611030100.

References

1. Chen,H.:Machine learning for information retrieval:neural networks, symbolic learning and genetic algorithms.Journal of the American Society for Information Science 46(1995)194–216.
2. Horng,J.T.,Yeh,C.C.:Applying genetic algorithms to query optimization in document retrieval.Information Processing and Management 36(2000)737–759.
3. Eberhart,R.C.,Kennedy,J.:A new optimizer using particle swarm theory. In:Proceedings of the Sixth International Symposium on Micro Machine and Human Science,Nagoya,Japan (1995)39–43.
4. Kennedy,J.:The particle swarm:social adaptation of knowledge. In: Proceedings of 1997 IEEE International Conference on Evolutionary Computation,Indianapolis (1997)303-308.
5. Salton G.,Wang A.,Yang C.S.:A vector space model for information retrieval. Journal of the American Society for Information Science 18(1975)613–620.
6. Singhal,A.,Buckley,C.,Mitra,M.:Pivoted document length normalisation.In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,Zurich,Switzerland (1996)21–29.
7. Harman,D.:Overview of the first text retrieval conference (TREC-1). In:Proceedings of the First Text Retrieval Conference, Gaitherburg,USA (1992)1–20.
8. Bartell,B.T.,Cottrell,G.W.,Belew,R.K.:Optimizing similarity using multi-query relevance feedback. Journal of the American Society for Information Science 49(1998)742-761.

Modular Ontologies - A Formal Investigation of Semantics and Expressivity

Jie Bao, Doina Caragea, and Vasant G. Honavar

Artificial Intelligence Research Laboratory,
Department of Computer Science
Iowa State University, Ames, IA 50011-1040, USA
{baojie, dcaragea, honavar}@cs.iastate.edu

Abstract. With the growing interest in modular ontology languages to address the need for collaborative development, integration, and use of ontologies on the Web, there is an urgent need for a common framework for comparing modular ontology language proposals on the basis of criteria such as their semantic soundness and expressive power. We introduce an Abstract Modular Ontology (AMO) language and offer precise definitions of semantic soundness such as localized semantics and exact reasoning, and expressivity requirements for modular ontology languages. We compare Distributed Description Logics (DDL), $\mathcal{E}$-connections, and Package-Based Description Logics (P-DL) with respect to these criteria. Our analysis suggests that by relaxing the strong domain disjointedness assumption adopted in DDL and $\mathcal{E}$-connection, as P-DL demonstrated, it is possible to overcome some known semantic difficulties and expressivity limitations of DDL and $\mathcal{E}$-Connections.

1 Introduction

Recently, there is a growing interest in modular ontology languages such as Distributed Description Logics (DDL) [4] and its syntax C-OWL[5], $\mathcal{E}$-connections [12,9], Fusion of Abstract Description Systems (FADS) [1], and Package-extended Description Logics (P-DL) [3]. Two broad classes of approaches are adopted to asserting and using semantic relations between multiple ontology modules: DDL and $\mathcal{E}$-connections adopt the "linking" approach that assumes that the modules are *nonoverlapping* or *disjoint*, while P-DL adopts the "importing" approach that allows direct use of foreign terms in an ontology module. Both DDL and P-DL cover scenarios that require inter-module concept subsumptions (e.g., *Dog* is *Animal*), while $\mathcal{E}$-connections allows only inter-module role relations (e.g., *DogOwner* owns *Dog*). Serafini *et.al.* (2005) [16] compared several mapping languages such as DDL and $\mathcal{E}$-connections, by reducing them to the Distributed First Order Logics (DFOL) [7] framework. Others have noted some of the semantic difficulties and limitations of such approaches [9,2].

However, there has been relatively little work on precise requirements for, and criteria for evaluating, modular ontology languages in more general settings that encompass both linking and importing among ontology modules. Some natural

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 616–631, 2006.
© Springer-Verlag Berlin Heidelberg 2006

questions that arise in comparing different approaches to integration of ontology modules are: What are the minimal requirements for ensuring the semantic soundness of a modular ontology language? What ontology language features are needed to construct a practical modular ontology? Under what circumstances can a reasoning process in a modular ontology language be said to be sound and complete? What are the sources of semantic difficulties in some modular ontology languages? How can such difficulties be avoided?

The goal of this paper is to provide some preliminary answers to these questions. Section 2 points out limitations of OWL to motivate the need for modular ontology languages. Section 3 explores a set of evaluation criteria for modular ontology languages. Section 4 precisely defines the aforementioned criteria within the Abstract Modular Ontology framework. Sections 5 and 6 (respectively) compare the semantic soundness and expressivity of several existing modular ontology language proposals w.r.t. the introduced criteria. 7 concludes with a summary and a brief discussion of related research.

2 Limitations of OWL as a Modular Ontology Language

OWL [15] is among the leading candidates for for a web ontology language. Hence, it is natural to ask why OWL cannot be used as a satisfactory modular ontology language.

We start by observing that OWL adopts an *importing* mechanism to support the integration of ontology modules. Thus, an OWL ontology may contain annotations `owl:imports` with references to other OWL ontologies. Once an OWL ontology O_1 imports another OWL ontology O_2, the terms defined in O_2 can be directly used in O_1 as *foreign terms*. In this manner, an ontology can be divided into smaller components within separate identification spaces, such as XML name spaces. However, the importing mechanism in OWL, in its current form, suffers from several serious drawbacks. In what follows, we will illustrate these drawbacks using a concrete example, the well-known Wine Ontology.

The wine ontology is given in two OWL files[1] focused on wine knowledge and general food knowledge, respectively. However, such a division into different files, a.k.a., XML name spaces:

- Does not support *localized semantics*. The inference is necessarily performed on the integrated centralized ontology of Wine and Food. The OWL semantics [14] requires that for any OWL ontology O and any abstract OWL interpretation I of O, "I satisfies each ontology mentioned in an owl:imports annotation directive of O". Therefore, it directly introduces both terms and axioms of the imported ontologies into the referring ontology (e.g., Food to Wine), which results in a global interpretation of all modules [4].
- Does not allow *local point of view*. All modules are required to adopt completely the same semantic perspective. For example, if the Food module as-

[1] http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine.rdf and http://www.w3.org/TR/2004/REC-owl-guide-20040210/food.rdf

serts "$Meat$ and $Fowl$ are disjoint", the Wine module cannot adopt another point of view asserting that "$Fowl$ is a type of $Meat$".

- Does not support *directional semantic relations.* Since a global model is used, the semantic constraints specified in a referring ontology (e.g., Wine) will be completely transfered over the imported ontologies (e.g., Food). If the Wine module adds "$Fowl$ is a type of $Meat$", the consistency of the Food module is also violated.
- Does not support *partial reuse.* Two modules mutually import each other, therefore a user has to import ALL the Wine and Food modules, even though only a small part (such as *Grape* classification) may be needed. Thus in order for an ontology module to be reused, it has to be either completely imported or completely discarded.

In conclusion, the current OWL importing mechanism [14] is only a syntactic solution, not a semantically sound solution for modular web ontologies.

3 Desiderata of Modular Ontologies

The observations in the previous section provide some intuitions that suggest evaluation criteria for modular ontology languages. The first set of criteria that we consider is aimed at evaluating the semantic soundness of such languages:

1. **Localized Semantics.** A modular ontology should not only be *syntactically modular* (e.g., stored in separated XML name spaces), but also *semantically modular.* That is, the existence of a *global model* should not be a requirement for integration of ontology modules.
2. **Exact Reasoning.** The answer to a reasoning problem over a collection of ontology modules should be *semantically equivalent* to that obtained by reasoning over an ontology resulting from an appropriate *integration* of the relevant ontology modules.
3. **Directional Semantic Relations.** The language must support *directional semantic relations* from a *source* module to a *target* module. A directional semantic relation affects only the reasoning within the target module and not the source module.
4. **Transitive Reusability.** Knowledge contained in ontology modules should be directly or indirectly reusable. That is, if a module Wine reuses module Food, and module Food reuses module Drink, then effectively, module Food reuses module Drink. If that is not the case, the knowledge in Food may be used in an unsafe or altered way. For example, if the Drink module contains "*Alcohol* is *Beverage*", the Food module contains "*Beer* is *Alcohol*" and "*Beverage* is *EdibleThing*", the Wine module may not be able to infer "*Beer* is *EdibleThing*" if knowledge in the Drink module is not considered.
5. **Decidability.** The ontology should be decidable, i.e. there is a decision procedure that can do reasoning on the modular ontology in finite time.

Other desiderata for sound semantics that have been considered in the literature include: the ability to cope with inconsistencies [4] and local logic completeness [10]. We believe that the criteria listed above are among the most critical ones for a modular ontology to be semantically sound and practically usable.

The second group of requirements that we consider is aimed at evaluating the language expressivity:

- **Concept Subsumption** (and its special case, **Concept Equivalency**) between modules is probably the most urgently needed feature. For example, the Wine module should be able to extend the Food module with $wine : WineGrape \sqsubseteq food : Grape$.
- **Concept Construction with Foreign Concepts.** It enables a module to build new local concepts based on concepts from other modules, using operators such as negation ($\neg$) , conjunction ($\sqcap$) and disjunction ($\sqcup$).
- **Concept Construction with Role Restrictions.** If R is a role and C is a concept (such that, one or both of them are foreign terms), the language may include existential restrictions ($\exists R.C$), universal restrictions ($\forall R.C$) and qualified number restrictions (e.g., $\leq 2R.C$).
- **Role Inclusion** (and its special case, **Role Equivalency**) between modules. For example, $wine : madeFromGrape \sqsubseteq food : madeFromFruit$.
- **Role Inversion** between modules. For example, $wine : madeIntoWine$ is the inverse of $food : madeFromFruit$.
- **Role Construction**, such as role complement ($\neg R$) , conjunction ($R \sqcap Q$) and disjunction ($R \sqcup Q$), where R and/or Q may be roles from different modules.
- **Transitive Role**, which allows the usage of a foreign transitive role, e.g., the Wine module reuses a transitive role *locatedIn* in the Region module.
- **Nominal Correspondence**. For example, the Wine module declares that the local individual CA is the same as the individual $California$ in the Region module.

Not all applications require the full expressive power of modular ontologies. Based on different intended application scenarios, a specific modular ontology language may only contain a proper subset of the given expressivity features. For example, DDL covers concept subsumption and nominal correspondence, and $\mathcal{E}$-connections addresses only concept and role construction with a special type of roles called “links”.

4 The Abstract Modular Ontology

This section studies an extended type of DFOL, an Abstract Modular Ontology (AMO) language, that will serve as the common testbed for investigating existing approaches according to the criteria introduced above.

4.1 An Abstract Modular Ontology Language

“Ontology is the science of being”(Aristotle, *Metaphysics*). In a general sense, a modular ontology is a set of individual descriptions of the same domain (e.g., Food) that represent correlated, but not identical points of view of multiple observers, or agents. Thus, each ontology module can be seen as describing a point of view held by an agent with respect to the entities (objects) and their relations in the domain. We say that a *domain relation* r_{ij} reflects the ability of an agent j to explain the point of view of an agent i, therefore it is a subjective *belief* rather than an objective *description*.

The idea presented here is strongly influenced by the Local Model Semantics [6] (which has also been influential on DFOL). However, instead of assuming a single relation between each pair of agents as in DFOL, we assume that each agent may need to interact with another agent throught different roles in different contexts. For example, a company can both buy and sell products from and to another company. Consequently, there may be multiple domain relations between ontologies held by a pair of agents.

A DFOL knowledge base (KB) [7] includes a family of first order languages $\{L_i\}_{i \in I}$, defined over a finite set of indices I. We will use L_i to refer to the ith module of the KB. An (i-)variable x or (i-)formula ϕ occurring in module L_i is denoted as $i : x$ or $i : \phi$ (we drop the prefix when there is no confusion). The signature (the set of all names) of L_i are i-terms. An **Abstract Modular Ontology** (AMO) is a DFOL KB in which each component language L_i is a subset of description logics (DL).

A model of AMO includes a set of local models and domain relations. For each L_i, there exists an interpretation domain Δ_i. Two domains Δ_i and Δ_j are *not necessarily* disjoint. Let M_i be the set of all DL models of L_i on Δ_i. We call each $m \in M_i$ a *local model* of L_i. A *domain relation* r_{ij}, where $i \neq j$, is a subset of $\Delta_i \times \Delta_j$. A domain relation r_{ij} represents the capability of the module j to map the objects of Δ_i into Δ_j. Each pair of local models may have multiple domain relations, each denoted by r_{ij}^{R} where R is the name for the domain relation. For any domain relation r_{ij}^{R}, we use $\langle d, d' \rangle \in r_{ij}^{R}$ to denote that from the point of view of j, the object d in Δ_i is mapped to the object d' in Δ_j, via relation R. In particular, a special domain relation $r_{ij}^{\rightarrow}$ (read as “image”) implies that the object d' in the j's point of view denotes the same entity as the object d in the i's point of view; d' is an image of d and d is a pre-image of d'. Note that the image relations, in general, are *not necessarily* one-to-one. Finally, $r_{ij}^{R}(d)$ denotes the set $\{d' \in \Delta_j | \langle d, d' \rangle \in r_{ij}^{R}\}$. For a subset $D \subseteq \Delta_i$, $r_{ij}^{R}(D)$ denotes $\cup_{d \in D} r_{ij}^{R}(d)$.

Example 1. *An ontology contains two modules $L_{\{1,2\}}$. L_1 contains knowledge about food objects and their relations, such as $Grape \sqsubseteq Fruit$ (a 1-formula). L_2 contains knowledge about wine objects and their relations, such as $Wine \sqsubseteq \forall madeFrom.Grape$. The local domain Δ_1 has Grape objects* ThompsonSeedless*,* CabernetFrancGrape*, and local domain Δ_2 has $Wine$ object* KathrynKennedyLateral *and Grape object* WineGrape*. The image domain relation $r_{21}^{\rightarrow}$ is $\langle 2 :$* WineGrape*$, 1 :$* CabernetFrancGrape*$\rangle$, while the image domain relation $r_{12}^{\rightarrow}$ is $\langle 1 :$* ThompsonSeedless*$, 2 :$* WineGrape*$\rangle, \langle 1 :$* CabernetFrancGrape*$, 2 :$* WineGrape*$\rangle$, and another domain relation $r_{12}^{madeWine}$ is $\langle 1 :$* CabernetFrancGrape*$, 2 :$* KathrynKennedyLateral*$\rangle$. $r_{12}^{\rightarrow}(1 :$* ThompsonSeedless*$) = \{2 :$* WineGrape*$\}$. Note* *that* $r_{\overrightarrow{21}} \neq (r_{\overrightarrow{12}})^{-}$ *since* L_1 *does not regard* 1 : *ThompsonSeedless as an image of* 2 : *WineGrape.*

4.2 Expressivity of Abstract Modular Ontology

Consider an agent j is observing the point of view of agent i and finds i uses x to identify an entity (e.g. a grape) in the world, which is identified by j as y. Therefore, j creates an image domain relation $i : x \rightarrow j : y$. If j finds that a set of objects in i's point of view is grouped as $Grape^{m_i}$ by i, then j will regard $r_{\overrightarrow{ij}}(Grape^{m_i})$ as "these objects in my point of view correspond to the *concept Grape* from agent i's point of view". Thus, j can also map *relations* in i's mind to its local point of view: for any relation instance $\langle x_1, x_2 \rangle$ in $\Delta_i \times \Delta_i$, j will regard $r_{\overrightarrow{ij}}(x_1) \times r_{\overrightarrow{ij}}(x_2)$ as a proper image of the relation. It should also be kept in mind that r_{ij} is always a relation viewed from j's point of view. For example, the fact that a person x thinks "y is my best friend" doesn't necessarily mean that y thinks "I'm x's best friend".

Therefore, the image of an i-concept C or i-role P in j is:

- $C^{i \rightarrow j}$: $r_{\overrightarrow{ij}}(C^{m_i})$
- $P^{i \rightarrow j}$: $\bigcup_{\langle x,y \rangle \in P^{m_i}} r_{\overrightarrow{ij}}(x) \times r_{\overrightarrow{ij}}(y)$

Similarly, pre-image of a j-concept D or a j-role R in i is defined as

- $D^{i \leftarrow j}$: $(r_{\overrightarrow{ij}})^{-}(D^{m_j})$
- $R^{i \leftarrow j}$: $\bigcup_{\langle x,y \rangle \in R^{m_j}} r_{\overrightarrow{ij}}^{-}(x) \times r_{\overrightarrow{ij}}^{-}(y)$

A concrete modular ontology language, such as DDL, $\mathcal{E}$-Connections or P-DL, usually contains a set of semantic relation rules, e.g. bridge rules (DDL), $\mathcal{E}$-connection, or concept importings (P-DL), between two ontology modules. Serafini et al. [16] have noted that such rules can be mapped to DFOL interpretation constraints in the form of $i : \phi(x_1, ..., x_n) \rightarrow j : \psi(y_1, ..., y_n)$, where ϕ, ψ are n-ary predicates and $\langle x_i, y_i \rangle$ is connected by a domain relation r_{ij}. Note that DL concepts are unary FOL predicates and DL roles are binary predicates. Consequently, a semantic relation in AMO will be either a concept inclusion axiom or a role inclusion axiom.

In a more general setting, a modular ontology language may also create third party constraints. For example, module j may reuse i-concept $RedWine$ and k-concept $Beverage$, and locally declare $i : RedWine \sqsubseteq k : Beverage$. However, such a third party constraint can be avoid by an "alias" syntax sugar such that $RedWine^{i \rightarrow j}$ and $Beverage^{k \rightarrow j}$ are given local alias $RedWine'$, $Beverage'$ thus transforming the concept inclusion to the one that connects only j-concepts.

A local concept can also be a complex concept constructed with a foreign role and/or a foreign concept, such as universal restriction (e.g.$\forall R.C$) or existential restriction (e.g.$\exists R.C$), as shown in the Table 1. However, arbitrary combination of the *possible* expressivity features in AMO may even lead to undecidability, since the union of multiple decidable logics may be undecidable[1]. The design of a practical modular ontology language has to be a tradeoff between the expressivity and reasoning complexity.

Table 1. Possible AMO Expressivity Features

	Syntax	**Semantics**
Concept Subsumption	$C \sqsubseteq D$	$C^{i \to j} \subseteq D^{m_j}$
	$C \sqsupseteq D$	$C^{i \to j} \supseteq D^{m_j}$
Concept Negation	$\neg C$	$r_{\overrightarrow{ij}}(\Delta_i \backslash C^{m_i})$
Concept Conjunction	$C \sqcap D$	$C^{i \to j} \cap D^{m_j}$
Concept Disjunction	$C \sqcup D$	$C^{i \to j} \cup D^{m_j}$
Universal Restriction	$\forall R.C$	$\{x \in \Delta_j \mid \forall y \in \Delta_i, (y, x) \in r_{ij}^{R} \to y \in C^{m_i}\}$
	$\forall P.D$	$\{x \in \Delta_j \mid \forall y \in \Delta_j, (x, y) \in P^{i \to j} \to y \in D^{m_j}\}$
	$\forall P.E$	$\{x \in \Delta_j \mid \forall y \in \Delta_k, \exists y' \in r_{\overrightarrow{kj}}(y) \wedge (x, y') \in P^{i \to j} \to y \in E^{m_k}\}$
Existential Restriction	$\exists R.C$	$\{x \in \Delta_j \mid \exists y \in \Delta_i, (y, x) \in r_{ij}^{R}, y \in C^{m_i}\}$
	$\exists P.D$	$\{x \in \Delta_j \mid \exists y \in \Delta_j, (x, y) \in P^{i \to j}, y \in D^{m_j}\}$
	$\exists P.E$	$\{x \in \Delta_j \mid \exists y \in \Delta_k, \exists y' \in r_{\overrightarrow{kj}}(y) \wedge (x, y') \in P^{i \to j}, y \in E^{m_k}\}$
Number Restriction[1]	$\leq nR.C$	$\{x \in \Delta_j \mid \#(\{y \in \Delta_i \mid (y, x) \in r_{ij}^{R}, y \in C^{m_i}\}) \leq n\}$
	$\leq nP.D$	$\{x \in \Delta_j \mid \#(\{y \in \Delta_j \mid (x, y) \in P^{i \to j}, y \in D^{m_j}\}) \leq n\}$
	$\leq nP.E$	$\{x \in \Delta_j \mid \#(\{y \in \Delta_k \mid \exists y' \in r_{\overrightarrow{kj}}(y) \wedge (x, y') \in P^{i \to j}, y \in E^{m_k}\}) \leq n\}$
Role Inclusion	$P \sqsubseteq R$	$P^{i \to j} \subseteq R^{m_j}$
	$P \sqsupseteq R$	$P^{i \to j} \supseteq R^{m_j}$
Role Inverse	P^-	$\{(y, x) \mid (x, y) \in P^{i \to j}\}$
Role Complement	$\neg P$	$(\Delta_j \times \Delta_j) \backslash P^{i \to j}$
Role Conjunction	$P \sqcap R$	$P^{i \to j} \cap R^{m_j}$
Role Disjunction	$P \sqcup R$	$P^{i \to j} \cup R^{m_j}$
Transitive Role	$trans(P)$	$(P^{i \to j})^+ = P^{i \to j}$
Nominal	$\{x\} \to \{y\}$	$y \in r_{\overrightarrow{ij}}(x)$

[1] $\geq$ case is similar.

C is an i-concept, D is a j-concept, E is a k-concept; P is an i-role, R is a j-role, Q is a k-role; x is a i-individual, y is a j-individual; $i \neq j$, $j \neq k$, i may be or may not be k. All formulas represent module j's point of view and constructed concepts (roles) are j-terms. Local domains of modules may be partially overlapping.

4.3 Semantic Soundness of the Abstract Modular Ontology

To precisely specify the semantic soundness of AMO, we need to answer several questions. First, what are the logical consequences in an AMO? How can local constraints in the agents' local points of view influence each other? For example, if agent i thinks "a is b's best friend", and agent j thinks $i : a$ is x and $i : b$ is y in j's mind, will j also hold the constraint that "x is y's best friend"?

Second, if there are inconsistencies in the points of view of two agents, what is the possible cause of such consistencies? For example, if agent j holds the belief that "x is y's enemy", possible causes can be either i and j hold incompatible points of view while the domain relations ($a \to x, b \to y$) are sound, or i and j actually hold compatible points of view but the domain relations are wrong (e.g. j has mistaken z as y and label both y and z as b locally, while z is x's enemy). While the first type of inconsistency is hard to eliminate (subjectivity), are there principled ways to avoid the second type of inconsistency (miscommunication)?

Third, if beliefs of agents are compatible, what is an "objective" way to integrate their knowledge? Or in other words, to "restore" a description of the physical world that reflects the consensus among the agents, such that logical consequences are consistent in the *integrated point of view* and each local point of view. For example, if a person *Alice* (identified as $i : a$ and $j : x$) behaves as the best friend of another person *Bob* (identified as $i : b$ and $j : y$), how can we construct an "integrated" description that is acceptable by both i and j, such that if $i(j)$ asserts a conclusion (e.g. x is y's best friend), the "integrated" description can also confirm the conclusion?

Addressing such problems is critical in identifying and solving several semantic difficulties that arise in modular ontology languages. Next, we introduce some definitions that are useful in precisely stating problems such as those we informally outlined above.

Definition 1 (AMO Satisfiability). *Let $M = \langle\{m_i\}, \{r_{ij}\}\rangle$ be a model for an AMO $O = \{L_i\}$ with interpretation constraint sets $\{C_{ij}\}$ (as defined in Table 1) , where $m_i = \langle\Delta_i, (.)_i\rangle$ is the local interpretation of i (Δ_i is the local domain of i, $(.)_i$ is the assignment function of i) and r_{ij} denotes all domain relations between m_i and m_j, including "image ($\rightarrow$)". We say that M satisfies O, denoted as $M \vDash O$, iff $m_i \vDash L_i$, for all i, and $M \vDash C_{ij}$, for all i and j.*

Definition 2 (AMO Entailment). *An AMO $O = \{L_i\}$ entails $C \sqsubseteq D$, where C, D are j-concepts, iff for any model $M = \langle\{m_i\}, \{r_{ij}\}\rangle$ of O, $m_j \vDash C \sqsubseteq D$.*

Although the above definition only addresses intra-module subsumption, it can be easily extended to inter-module subsumption with a simple syntax rewriting. If C is an i-concept, we can always create a j-concept C' interpreted as $C^{i\rightarrow j}$, and then $i : C \sqsubseteq j : D$ can be transformed as $j : C' \sqsubseteq j : D$.

Definition 3 (Localized and Globalized Semantics). *An AMO $O = \{L_i\}$ has only globalized semantics, iff for any model $M = \langle\{m_i\}, \{r_{ij}\}\rangle$ of O, $M \vDash O$, $m_i = \langle\Delta_i, (.)_i\rangle$, local domains $\{\Delta_i\}$ of $\{L_i\}$ must be identical. Otherwise, it has localized semantics.*

Definition 4 (Decidability). *An AMO $O = \{L_i\}$ is decidable if for every satisfiability problem (therefore also entailment problem) C for i, there exists an algorithm that is capable of deciding in a finite number of steps whether there exists a model $M = \langle\{m_i\}, \{r_{ij}\}\rangle$, $M \vDash O$, such that C is satisfiable in m_i.*

Definition 5 (Directional Semantic Relations). *Domain relations in an AMO are directional, iff for any model $M = \langle\{m_i\}, \{r_{ij}\}\rangle$ of O, for any $i \neq j$, $m_j \vDash C \sqsubseteq D$, doesn't imply that $C^{i\leftarrow j} \subseteq D^{i\leftarrow j}$ must be true in m_i.*

Transitive reusability means that an agent can infer local constraints based on observing constraints in other agents' points of view. For example, if i believes "a is b's best friend", and j believes domain relation $i : a \rightarrow j : x, i : b \rightarrow j : y$, then j may reuse i's knowledge and infer that "x is y's best friend". Furthermore, if another agent k who is confident in j's judgement, and believes $j : x \rightarrow k : p, j : y \rightarrow k : q$, then k also believes "p is q's best friend".

Definition 6 (Transitive Reusability). *For an AMO $O = \{L_i\}$, L_i is said to be* reusable *by j ($j \neq i$) if for any concepts C, D in L_i, such that $L_i \vDash C \sqsubseteq D$,* *we have that for $M = \langle\{m_i\}, \{r_{ij}\}\rangle$ of O, $C^{i\to j} \subseteq D^{i\to j}$ must be true in m_j. L_i is said to be* transitively reusable *if for any j, k ($i \neq j \neq k$), if L_i is reusable by L_j, and L_j is reusable by L_k, then we must have L_i is reusable by L_k.*

Exact reasoning means that the points of view of all agents can be reconciled into a point of view (a consensus) that is consistent with each individual agent's point of view. Since such a merged state will be the consensus of individual agents, their compatible beliefs may be combined. For example, if i believes x is the identifier of a person *Alice*, j believes a is the identifier of *Alice* and $i : x \to j : a$, then the merged state of the two agents will "believe" $i : x$ and $j : a$ are all identifiers of *Alice*. Thus, all semantic relation rules, in their DFOL form $i : \phi(x_1, ..., x_n) \to j : \psi(y_1, ..., y_n)$ (n=1 or 2), where x_i, y_i are connected by $r_{\overrightarrow{ij}}$, will be reduced to $\phi(x_1, ..., x_n) \to \psi(x_1, ..., x_n)$.

Definition 7 (Exact Reasoning). *Reasoning in an AMO $O = \{L_i\}$ is exact, iff for any model $M = \langle\{m_i\}, \{r_{ij}\}\rangle$ of O, there exists a classical model $M' = \Re(M) = \langle\Delta_m, (.)_m\rangle$, such that $M \vDash \phi \Rightarrow M' \vDash \phi$. $\Re$ denotes the reduction from M to M' as follows:*

- $\Delta_m = \cup_i \Delta_i$
- *The assignment function $(.)_m$ is defined as: for any concept $i : C$, $C^m = C^{m_i}$; for any role $i : P$, $P^m = P^{m_i}$; for any individual $i : I$, $I^m = I^{m_i}$.*
- *for every image domain relation, if $(i : x, j : y) \in r_{\overrightarrow{ij}}$, add $i : x = j : y$.*
- *for every other domain relation R, if $(i : x, j : y) \in r^R_{ij}$, assign (x, y) to R^m.*

5 Semantic Soundness of Existing Approaches

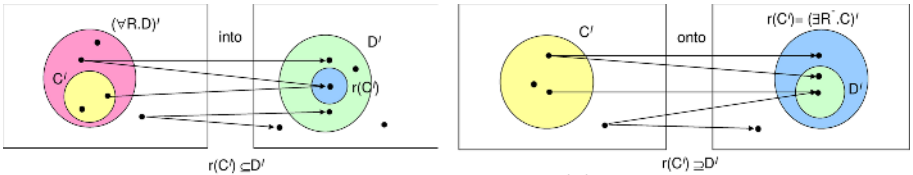
5.1 Distributed Description Logics

Distributed Description Logics (DDL) [4], adopts a "linking"-based approach. In DDL, the semantic mappings between disjoint modules L_i and L_j are established by a set of "Bridge Rules" (B_{ij}) of the form:

- INTO rule: $i : \phi \xrightarrow{\sqsubseteq} j : \psi$, semantics: $r_{ij}(\phi^{m_i}) \subseteq \psi^{m_j}$
- ONTO rule: $i : \phi \xrightarrow{\sqsupseteq} j : \psi$, semantics: $r_{ij}(\phi^{m_i}) \supseteq \psi^{m_j}$
- Individual Correspondence: $i : a \to j : b$, semantics: $b^{m_j} \in r_{ij}(a^{m_i})$

where $m_i(m_j)$ is a model of $L_i(L_j)$, ϕ, ψ are formulae; r_{ij} is a domain relation which serves as the interpretation of B_{ij}, and can be seen as the image domain relation $r_{\overrightarrow{ij}}$ in AMO. Although ϕ, ψ may be role names[5,16], semantics and decidability of such an extension is still not well-understood. The semantics of bridge rules between concepts is shown in Figure 1.

Distributed concept correspondence between two modules in DDL covers some of the most important scenarios that require mapping between ontology modules. Since DDL has clear DFOL interpretation, it is easy to see that it has localized semantics and supports directional semantic relations. DDL is decidable if each connected module is decidable [4,12].



(a) Into Bridge Rules (b) Onto Bridge Rules

Fig. 1. Semantics of DDL Bridge Rules between Concepts

However, DDL, as noted in [9,8], doesn't ensure transitive reusability and exact reasoning: (a) *Subsumption Propagation problem*: concept subsumption links in DDLs do not propagate transitively. For example, in the case of 3 ontology modules $L_{\{1,2,3\}}$, the bridge rules $1 : Bird \xrightarrow{\sqsupseteq} 2 : Fowl$ and $2 : Fowl \xrightarrow{\sqsupseteq} 3 : Chicken$ do not in general ensure that $1 : Bird \xrightarrow{\sqsupseteq} 3 : Chicken$; (b) *Inter-module Unsatisfiability problem*: DDLs may not detect unsatisfiability across ontology modules. For example, $1 : Bird \xrightarrow{\sqsupseteq} 2 : Penguin$ and $1 : \neg Fly \xrightarrow{\sqsupseteq} 2 : Penguin$ do not render $2 : Penguin$ unsatisfiable even if L_1 entails $Bird \sqsubseteq Fly$.

A primary source of such difficulties has to do with the fact that the domain relations in DDL can be arbitrary [2]. In the absence of a formal mechanism to prevent inconsistency between the agents' points of view due to miscommunication, domain relations cannot be reused by other modules (r_{13} cannot be inferred from r_{12} and r_{23}, as illustrated by example (a) above). This precludes transitive reusability. Furthermore, unsatisfiability across ontology modules can not be detected (as illustrated by example (b) above) since objects of disjoint i-concepts can be mapped to the same object in j. Bao et.al. [2] have recently shown that one-to-one domain relation is a sufficient condition for exact DDL reasoning. However, at present, there is no principled approach to coming up with such domain relations in DDL alone.

5.2 $\mathcal{E}$-Connections

While DDL allows only one type of domain relations, the $\mathcal{E}$-connection approach allows multiple "link" relations between two domains. $\mathcal{E}$-connections between DLs [11,9] restrict the local domains of the $\mathcal{E}$-connected ontology modules to be disjoint (therefore ensure localized semantics). Roles are divided into disjoint sets of *local roles* (connecting concepts in one module) and *links* (connecting inter-module concepts). Formally, given ontology modules $\{L_i\}$, an (one-way binary) link $E \in \mathcal{E}_{ij}$, where $\mathcal{E}_{ij}, i \neq j$ is the set of all links from the module i to the module j, can be used to construct a concept in module i, with the syntax and semantics specified as follows:

- $\langle E\rangle(j : C)$ or $\exists E.(j : C) : \{x \in \Delta_i | \exists y \in \Delta_j, (x, y) \in E^M, y \in C^M\}$
- $\forall E.(j : C) : \{x \in \Delta_i | \forall y \in \Delta_j, (x, y) \in E^M \to y \in C^M\}\}$
- $\leq nE.(j : C) : \{x \in \Delta_i | \#(\{y \in \Delta_j | (x, y) \in E^M, y \in C^M\}) \leq n\}$
- $\geq nE.(j : C) : \{x \in \Delta_i | \#(\{y \in \Delta_j | (x, y) \in E^M, y \in C^M\}) \geq n\}$

where $M = \langle \{m_i\}, \{E^M\}_{E \in \mathcal{E}_{ij}} \rangle$ is a model of the $\mathcal{E}$-connected ontology, m_i is the local model of L_i; C is a concept in L_j, with interpretation $C^M = C^{m_j}$; $E^M \subseteq \Delta_i \times \Delta_j$ is the interpretation of a $\mathcal{E}$-connection E.

An advantage of $\mathcal{E}$-connections is that a collection of $\mathcal{E}$-connected ontology modules is decidable if all modules are decidable [12]. However, since there are no image domain relations in $\mathcal{E}$-connections, transitive reusability cannot be guaranteed in general. Some scenarios may still allow knowledge propagation. For example, if module i contains $D \sqsubseteq E$, module j contains $A \equiv \forall R.D$, $B \equiv \forall R.E$, where R is a $\mathcal{E}$-connection from j to i, j can infer that $A \sqsubseteq B$ must be true. $\mathcal{E}$-connections are also directional.

The exactness of reasoning (as defined in Definition 7) of $\mathcal{E}$-connections given in [9] can be guaranteed since there is no image domain relation. A reduction from a $\mathcal{E}$-connections model to a classical model can be obtained by constructing a simple union of all local models where all link instances are converted into classic role instances. However such a reduction does not hold in the case of "generalized links" [13] where a link/role name can be used within different contexts. For example, given two modules $L_{\{1,2\}}$, L_2 contains $Penguin \sqsubseteq \forall isa^{(1)}.(1 : Bird)$ and $Penguin \sqsubseteq \exists isa^{(2)}.(2 : PolarAnimal)$, where isa is interpreted as link or local role under different contexts. Since $1 : Bird$ and $2 : PolarAnimal$ are disjoint by default, the disjoint union of isa interpretation in each of its contexts will be unsatisfiable.

5.3 Package-Based Description Logics

Package-based Description Logics (P-DL)[3] offer a tradeoff between the strong module disjointness assumption of DDL and $\mathcal{E}$-connections, and on the other hand, the OWL importing mechanics, which forces *complete overlapping* of modules. In P-DL, an ontology is composed of a collection of modules called *packages*. Each term (name of a concept, a property or an individual) and each axiom is associated with a *home package*. A package can use terms defined in other packages i.e., *foreign terms*. If a package L_j uses a term $i : t$ with home package L_i $(i \neq j)$, then we say t is *imported* into L_j, and the importing relation is denoted as r^t_{ij}. In what follows, we will examine a restricted type of package extension which only allows import of concept names.

The semantics of P-DL is expressed in AMO as follows: For a package-based ontology $\langle \{L_i\}, \{r^t_{ij}\}_{i \neq j} \rangle$, a distributed model is $M = \langle \{m_i\}, \{(r_{\overrightarrow{ij}})^t\}_{i \neq j} \rangle$, where m_i is the local model of module i, $(r_{\overrightarrow{ij}})^t \subseteq \Delta_i \times \Delta_j$ is the interpretation for the importing relation r^t_{ij}, which meets the following requirements:

- Every importing relation is one-to-one in that it maps each object of t^{m_i} to a single unique object in t^{m_j}, therefore $(r_{\overrightarrow{ij}})^t(t^{m_i}) = t^{m_j}$.
- Term Consistency: importing relations of different terms are consistent, i.e., for any $i : t_1 \neq i : t_2$ and any $x, x_1, x_2 \in \Delta_i$, $(r_{\overrightarrow{ij}})^{t_1}(x) = (r_{\overrightarrow{ij}})^{t_2}(x)$ and $(r_{\overrightarrow{ij}})^{t_1}(x_1) = (r_{\overrightarrow{ij}})^{t_2}(x_2) \neq \emptyset \rightarrow x_1 = x_2$.
- Compositional Consistency: if $(r_{\overrightarrow{ik}})^{i:t_1}(x) = y_1$, $(r_{\overrightarrow{ij}})^{i:t_2}(x) = y_2$,$(r_{\overrightarrow{jk}})^{j:t_3}$ $(y_2) = y_3$, , (where t_1 and t_2 may or may not be same), and y_1, y_2, y_3 are

not null, then $y_1 = y_3$. Compositional consistency helps ensure that the transitive reusability property holds for P-DL.

The *image domain relation* between m_i and m_j is $r_{\overrightarrow{ij}} = \cup_t (r_{\overrightarrow{ij}})^t$ and is strictly one-to-one. From the multi-agent point of view, such a domain relation ensures unambiguous communication between each modules. Consequently, r_{ij} in a P-DL model isomorphically "copies" the relevant partial domain from m_i to m_j (Figure 2). Since the construction of a local model is dependent on the structure of local models of imported modules, P-DL allows a relaxation of the domain disjointedness assumption adopted in DDL and $\mathcal{E}$-connections. However, the loss of disjointedness does not sacrifice *localized semantics* property of modules, since they are, unlike in OWL, only partially overlapping. Consequently, there is no required global model.

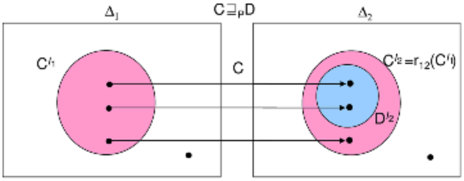


Fig. 2. Semantics of P-DL

With a principled way to avoid semantic imprecision, P-DL can ensure transitive reusability and exact reasoning [2]. A reduction from a P-DL model to a classical DL model is the union of all local models with"copied" objects being merged. However, semantic relations in P-DL may not always be directional in local domains that overlap: if module j imports concept C, D from i, then $C \sqsubseteq D$ in j will imply $C^{i \leftarrow j} \sqsubseteq D^{i \leftarrow j}$ in Δ_i.

The general decidability transfer property does not always hold in P-DL since the union of two decidable fragments of DL may be undecidable [1]. This presents semantic difficulties in the general setting of connecting ADSs [12]. However, in a setting where different ontology modules are specified using subsets of the *same* decidable DL language, such as $\mathcal{SHOIQ}(D)$ (OWL-DL), and importing is only allowed for concept names, the union of such modules is decidable. In such a setting, semantics-preserving reduction from P-DL model to the integrated DL model is available [2] making P-DL decidable.

The comparison is summarized in Table 2.

Table 2. Comparison of Semantic Soundness

	Localized Semantics	**Exact Reasoning**	**Directional Relation**	**Transitive Reusability**	**Decidability***
DDL	Yes	No	Yes	No	Yes^1
$\mathcal{E}$-Connections	Yes	Partial^2	Yes	Partial	Yes
OWL	No	Yes	No	Yes	Yes (OWL-DL)
P-DL	Yes	Yes	Partial	Yes	Partial^3

* when each local module is decidable; ^{1} yes only for concept bridge rules; ^{2} yes without generalized links; ^{3} yes for concept importing and each module from a subset of $\mathcal{SHOIQ}(D)$.

Table 3. Comparison of Expressivity of DDL, $\mathcal{E}$-connections and p-DL

	Syntax	**DDL**	**$\mathcal{E}$-Connections**	**P-DL**
Concept Subsumption	$C \sqsubseteq D$	$\surd$	$\times$	$\surd$
	$C \sqsupseteq D$	$\surd$	$\times$	$\surd$
Concept Negation	$\neg C$	$\surd$	$\times$	$\surd$
Concept Conjunction	$C \sqcap D$	$\surd$	$\times$	$\surd$
Concept Disjunction	$C \sqcup D$	$\surd$	$\times$	$\surd$
Universal Restriction	$\forall R.C$	$\times$	$\surd$	$\surd$
	$\forall P.D$	$\times$	$\times$	$\times$
	$\forall P.E$	$\times$	$\times$	$\times$
Existential Restriction	$\exists R.C$	$\times$	$\surd$	$\surd$
	$\exists P.D$	$\times$	$\times$	$\times$
	$\exists P.E$	$\times$	$\times$	$\times$
Number Restriction[1]	$\leq nR.C$	$\times$	$\surd$	$\surd$
	$\leq nP.D$	$\times$	$\times$	$\times$
	$\leq nP.E$	$\times$	$\times$	$\times$
Role Inclusion	$P \sqsubseteq R$	$\surd$	$\times$	$\times$
	$P \sqsupseteq R$	$\surd$	$\times$	$\times$
Role Inverse	P^-	$\times$	$\times$	$\times$
Role Complement	$\neg P$	$\times$	$\times$	$\times$
Role Conjunction	$P \sqcap R$	$\times$	$\times$	$\times$
Role Disjunction	$P \sqcup R$	$\times$	$\times$	$\times$
Transitive Role	$trans(P)$	$\times$	$\surd$[2]	$\times$
Nominal	$\{x\} \to \{y\}$	$\surd$	$\times$	$\times$

[1] $\geq$ case is similar. [2] only with generalized links.
Notations are as the same in Table 1. All formulas represent module j's point of view.

6 Expressivity of Existing Approaches

Table 3 shows the expressivity comparison of DDL, $\mathcal{E}$-connections and P-DL.

6.1 Distributed Description Logics

DDL bridge rules cannot be directly read as inter-module concept subsumptions. However, several techniques have been studied to simulate concept subsumptions with bridge rules [17]. DDL is also capable of using foreign concepts in local concept negation, conjunction and disjunction by a simple “alias” syntax sugar as mentioned in section 4.

However, major limitations in the expressivity of DDL have to do with linking modules with roles. Since DDL semantics only allows one type of domain relations, role instances cannot be created between local models. This precludes concepts built with foreign role or inter-module role construction. Although the extended DDL in [5,16] allows role inclusions, the decidability of such an extension as well reasoning algorithms that work in such a setting are still unknown.

6.2 $\mathcal{E}$-Connections

In contrast with DDL, $\mathcal{E}$-connections allow role connections between modules but doesn't allow inter-module concept inclusion. Although it has been argued that $\mathcal{E}$-connections are more expressive than DDLs [12,8], the intended use of DDL bridge rules and $\mathcal{E}$-connection links are quite different. This is made clear in the AMO framework, where DDL bridge rules are interpreted as image domain relations, and $\mathcal{E}$-connection links are not image domain relations. Since $\mathcal{E}$-connections strictly require local domain disjointedness, no direct inter-module concept subsumption can be allowed. Therefore, DDL bridge rules and $\mathcal{E}$-connection links actually cover different application scenarios, and thus are complementary in their roles.

It should be noted that the direction of "links" in $\mathcal{E}$-connections is the inverse of AMO roles defined in Table 1 and 3. In $\mathcal{E}$-connections, module i can use a link from i to j to construct an i-concept, whereas in AMO, i can only construct i-concepts with a role from j to i. This difference arises from AMO's underlying assumption that any domain relation r_{ij}^R is only a subjective point of view of j and should only be used in j. Therefore, we contend that a $\mathcal{E}$-connections link used in module i, although is syntactically given as from i to j, stands for the subjective point of view of i, not j. The difference can be syntactically eliminated by inverting E-connection links to obtain AMO roles.

However, the expressivity of $\mathcal{E}$-connections is limited by the need to ensure the disjointedness of local domains. Thus, a concept cannot be declared as a subclass of a foreign concept, and foreign concepts cannot be used in local concept constructions. A property cannot be declared as sub-relation of a foreign property, and neither foreign classes nor foreign properties can be instantiated. It is also difficult to combine $\mathcal{E}$-connections and OWL importing [8].

$\mathcal{E}$-connection links cannot be seen as foreign roles in AMO. Their usage is equivalent to allowing an AMO local role to have foreign concepts within its range. In the case of the generalized link property [13], the boundary between local roles and links is further ambiguous. The $\mathcal{E}$-connections syntax proposal [8] requires that the source of a link be the module in which it has been declared. Therefore, link constructors, such as inverse, conjunction, disjunction and complement, are different from the inter-module role constructors defined in Table 1, and are closer to intra-module role constructors.

6.3 Package-Based Description Logics

P-DL expressivity features summarized in table 3 only allow concept name importing, since decidability of more expressive variants of P-DL is still unknown. Despite its stronger domain relation restrictions (ont-to-one), P-DL is more expressive than DDL and $\mathcal{E}$-Connections in several ways. P-DL allows inter-module concept subsumption, concept construction with foreign concepts, and connecting modules with roles, thus provides several expressivity features that are missing either in DDL or $\mathcal{E}$-connections.

DDL with only concept correspondence and $\mathcal{E}$-connections without generalized links can be reduced to P-DL. For example, an into rule $i : C \xrightarrow{\sqsubseteq} j : D$ in DDL can be reduced to a P-DL axiom $C \sqsubseteq D$ in module j and C is an imported concept; A $\mathcal{E}$-connection-like constructed concept such as $\exists(i : E).(j : D)$ can be defined in the module i, where $j : D$ is imported into i, with semantics given Table 1; $\forall(i : E).(j : D)$ can be constructed in a similar fashion.

Therefore, we believe that the importing approach adopted by P-DL which relaxes the strong module disjointedness assumption of DDL offers the possibility of avoiding many of the semantic difficulties of current modular ontology language proposals while improving the expressivity.

7 Conclusions

This paper provides a formal investigation of the motivation, evaluation criteria, an abstract framework of modular ontologies, and compares the semantic soundness and expressivity of several modular ontology languages. The main contributions of this paper are: a) identification and precise definition of possible requirements for semantically sound modular ontology languages; b) identification of desirable expressivity features of modular ontology languages; c) introduction of an Abstract Modular Ontology (AMO) framework which offers a basis for comparing different modular ontology languages; d) comparison of the semantic soundness and the expressivity of DDL, $\mathcal{E}$-connections and P-DL; and e) analysis of several semantic difficulties and expressivity limitations of DDL and $\mathcal{E}$-connections, and propose an approach in the form of a partial importing mechanism in P-DL to overcome such limitations.

We conclude that different existing modular ontology language proposals are motivated by, and hence are responsive to, different application scenarios. At present, there is no modular ontology language with known decidability and inference complexity that supports both general inter-module concept and inter-module role correspondence and satisfies all semantic soundness requirement. Our results suggest that in order to improve the expressivity of existing modular ontology languages, and to ensure their semantic soundness, the strict module disjointedness assumption adopted by DDL and $\mathcal{E}$-connections may need to be at least partially relaxed. Work in progress is aimed at the development of a reasoning algorithm for an expressive and semantically sound modular ontology language, e.g. P-DL.

References

1. F. Baader, C. Lutz, H. Sturm, and F. Wolter. Fusions of description logics. In *Description Logics*, pages 21–30, 2000.
2. J. Bao, D. Caragea, and V. Honavar. On the semantics of linking and importing in modular ontologies (extended version). Technical report, TR-408 Computer Sicence, Iowa State University, 2006.

3. J. Bao, D. Caragea, and V. Honavar. Towards collaborative environments for ontology construction and sharing. In *International Symposium on Collaborative Technologies and Systems (CTS 2006)*, pages 99–108. IEEE Press, 2006.
4. A. Borgida and L. Serafini. Distributed description logics: Directed domain correspondences in federated information sources. In *CoopIS/DOA/ODBASE*, pages 36–53, 2002.
5. P. Bouquet, F. Giunchiglia, and F. van Harmelen. C-OWL: Contextualizing ontologies. In *Second International Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*, pages 164–179. Springer Verlag, 2003.
6. C. Ghidini and F. Giunchiglia. Local model semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.
7. C. Ghidini and L. Serafini. *Frontiers Of Combining Systems 2, Studies in Logic and Computation*, chapter Distributed First Order Logics, pages 121–140. Research Studies Press, 1998.
8. B. C. Grau. *Combination and Integration of Ontologies on the Semantic Web*. PhD thesis, Dpto. de Informatica, Universitat de Valencia, Spain, 2005.
9. B. C. Grau, B. Parsia, and E. Sirin. Working with multiple ontologies on the semantic web. In *International Semantic Web Conference*, pages 620–634, 2004.
10. B. C. Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Modularity and web ontologies. In *KR2006*, 2006.
11. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of description logics. In *Description Logics Workshop, CEUR-WS Vol 81*, 2003.
12. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of abstract description systems. *Artif. Intell.*, 156(1):1–73, 2004.
13. B. Parsia and B. C. Grau. Generalized link properties for expressive epsilon-connections of description logics. In *AAAI*, pages 657–662, 2005.
14. P. Patel-Schneider, P.Hayes, and I. Horrocks. Web ontlogy language (owl) abstract syntax and semantics. http://www.w3.org/TR/owl-semantics/, February 2003.
15. G. Schreiber and M. Dean. Owl web ontology language reference. http://www.w3.org/TR/2004/REC-owl-ref-20040210/, February 2004.
16. L. Serafini, H. Stuckenschmidt, and H. Wache. A formal investigation of mapping language for terminological knowledge. In *IJCAI*, pages 576–581, 2005.
17. L. Serafini and A. Tamilin. Drago: Distributed reasoning architecture for the semantic web. In *ESWC*, pages 361–376, 2005.

A Pi-Calculus Based Ontology Change Management[⋆]

Meiling Wang and Lei Liu[⋆⋆]

Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education of P.R. China, College of Computer Science and Technology, Jilin University, Changchun, 130012, P.R. China
liulei@jlu.edu.cn

Abstract. Based on the pi-calculus, this paper proposes a kind of ontology process model used for solving the change implementation and propagation problems of ontology evolution process. This solution is discussed at three levels: the change implementation of single ontology evolution, the push-based synchronization realization for the change propagation in the evolution of multiple dependent ontologies within a single node, and the pull-based synchronization realization for the change propagation of the distributed ontologies evolution.

1 Introduction

In a more open and dynamic environment, due to the changes in the application's domain or the user's requirements, the domain knowledge changes over time and ontology evolves continually [1]. A modification in one part of an ontology may generate some subtle inconsistencies in the other parts of the same ontology, in the ontology-based instances as well as in the dependent ontologies and applications [2]. Ontology evolution is the timely adaptation of an ontology to the arisen changes and the consistent propagation of these changes to dependent artefacts [3], and a six-phase evolution process is proposed in [4].

Pi-calculus is a kind of calculus of "mobile" processes and is used to model concurrent and dynamic systems [5]. Based on it, this paper proposes a kind of ontology process model and discusses the change implementation and propagation problems of ontology evolution process. Section 2 gives a brief introduction to pi-calculus. Section 3 describes the ontology process model. Section 4 elaborates respectively on managing changes at three levels. Section 5 is an overview of related work. Section 6 concludes the paper and discusses some further work.

2 An Overview of Pi-Calculus

The basic computing entities of pi-calculus are names and processes, and the communication between two processes is realized by transferring objects along

[⋆] This paper is sponsored by European Commission under grant No.TH/Asia Link/010 (111084) and Jilin province science development plain project of China under grant No. 20050527.

[⋆⋆] Corresponding author.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 632–638, 2006.
© Springer-Verlag Berlin Heidelberg 2006

their link (port). Link names belong to the same category as the transferred objects, thus the link name between two processes can be transferred so as to change interconnections as they interact.

A simple communication between P and Q is shown in Fig.1. P and Q are processes, x, y and z are link (port) names; P sends a message along $\overline{y}$, and Q receives the message from y. The main syntax of pi-calculus is described in [6].

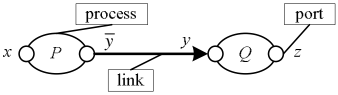


Fig. 1. Communication between P and Q

3 Ontology Process Model

Based on the pi-calculus, this paper proposes a kind of ontology process model. Entities of a concrete ontology including concepts, properties and instances are presented as processes; associations between entities are denoted as the links between processes, and an entity can interoperate with another one associated with it by the link between them; and the ontology itself can be denoted as a complex process equal to the parallel composition of all the entity processes contained. Fig.2 is a simple ontology example, whose process model is shown in Fig.3 and is described as:$Root|Person|Project|Works\_at|Prof|Student|PhD|MSc|WiHi|ResearchProject|Ontologging$.

4 Managing Changes Using Pi-Calculus

Based on the ontology process model described above, this section will solve the implementation and propagation problems of the changes in ontology evolution process using pi-calculus, especially three aspects are discussed as follows.

4.1 Single Ontology Process Model Evolution

Three elementary change operations shown in Fig.4: *CreateEntity*, *DeleteEntity* and *ModifyEntity* are defined as the processes as follows:

CreateEntity$(E_1, E_2, x) \overset{def}{=} \overline{x}(CREATE).x(msg_1).[msg_1 = CREATEACK]\overline{x}(END).E_2|x(msg_2).[msg_2 = CREATE]\overline{x}(CREATEACK).E_1$

DeleteEntity$(E_1, E_2, x) \overset{def}{=} \overline{x}(DELETE).x(msg_1).[msg_1 = DELETEACK]0|x(msg_2).[msg_2 = DELETE]\overline{x}(DELETEACK).E_1$

ModifyEntity$(E_1, E_2, E_3, x, y) \overset{def}{=} \overline{x}(MODIFY).x(msg_1).[msg_1{=}MODIFYACK]\overline{x}(END).E_2|x(msg_2).[msg_2 = MODIFY]\overline{y}(MODIFY).y(msg_3).[msg_3 = MODIFYACK]\overline{x}(MODIFYACK).\overline{y}(x).E_1|y(msg_4).[msg_4{=}MODIFY]\overline{y}(MODIFYACK).$

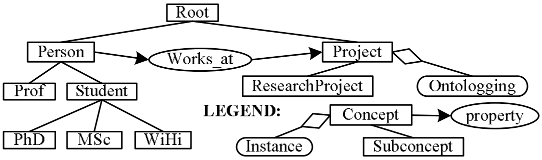


Fig. 2. A simple ontology

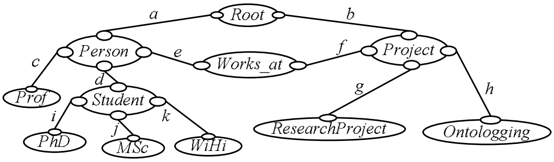


Fig. 3. An ontology process model

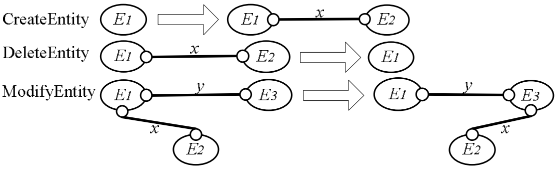


Fig. 4. CreateEntity,DeleteEntity and ModifyEntity

$y(z).E_3$ Message names in the set $\{CREATE, CREATEACK, DELETE, DELETEACK, MODIFY, MODIFYACK, BEGIN, END\}$are used for the synchronization of change operations, for example *CREATE* and *CREATEACK* are for *CreateEntity*. Complex change operation is defined as the composition of elementary operations, and *BEGIN* and *END* are for the composition. *CreateEntity* takes precedence over *ModifyEntity*, and *ModifyEntity* takes precedence over *DeleteEntity*.

For single ontology evolution process [7], the essential phase is the *semantics of change*, whose task is to maintain ontology consistency, and in which an *evolution strategy* which unambiguously defines the way how an ontology change will be resolved resulting in a consistent state fulfilling the user's preferences [4], is typically chosen by the user at the start of the process. Assume that for the concept removal to reconnect subconcepts to the parent concepts, then the removal of *Student* from Fig.3 will be implemented as

$\overline{d}(BEGIN).(i(msg_1)|j(msg_2)|k(msg_3)).[msg_1 = END][msg_2 = END][msg_3 = END]$ $DeleteEntity(Person, Student, d)|d(msg).[msg = BEGIN](ModifyEntity(Student, PhD, Person, i, d)|ModifyEntity(Student, MSc, Person, j, d)|ModifyEntity(Studen - t, WiHi, Person, k, d))$, where the reconnection from *PhD*, *MSc* and *WiHi* to *Person* must take precedence over the deletion of *Student*.

4.2 Evolution of Multiple Dependent Ontology Process Models

A dependent ontology is consistent if the ontology itself and all its included ontologies, observed alone and independently of the ontologies in which they are reused, are single ontology consistent [8]. Fig.5 presents four ontology process models: *SO*, *BO*, *CO*, and *ICO*. *BO* and *CO* each include *SO* and *ICO* includes *BO* and *CO*, which is shown on the right-hand side. If *Sports Utility* of *SO* is deleted, *BO* and *ICO* will become inconsistent since *Bicycle* and *Catalog Item* will have a superconcept and a subconcept undefined respectively.

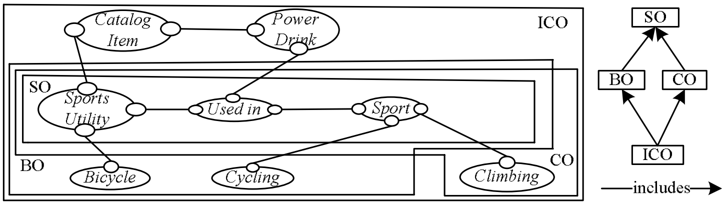


Fig. 5. Four Dependent Ontologies

Consider the inclusion relationships among the ontologies, except for cyclical inclusions and subsets inclusions. The consistency maintenance of multiple dependent ontologies within one node may be achieved by the *push-based* synchronization approach: changes of the changed ontology are propagated to dependent ontologies as they happen [9]. To avoid temporal inconsistency, changes should be pushed immediately as they occur [10]. For Fig.5, the propagation order is $SO \rightarrow BO|CO \rightarrow ICO$; the links between *BO* and *SO*, *CO* and *SO*, *BO* and *ICO*, *CO* and *ICO* are x, y, z and t respectively; if *SO* is changed and the operation is *action*, then the *push-based* synchronization will be realized as

$$action.action_1.(\overline{x}(BEGIN)|\overline{y}(BEGIN)).SO|x(msg_1).[msg_1 = BEGIN]action_2.\overline{z}(BEGIN).BO|y(msg_2).[msg_2 = BEGIN]action_3.\overline{t}(BEGIN).CO|(z(msg_3).[msg_3 = BEGIN]action_4|t(msg_4).[msg_4 = BEGIN]action_5).action_6.\overline{t}(END).ICO,$$

where $action_1$, $action_2$, $action_3$ and $action_6$ are generated by *action* on *SO*, *BO*, *CO* and *ICO* respectively; $action_4$ is generated by $action_2$ on *ICO*; and $action_5$ is generated by $action_3$ on *ICO*.

4.3 Evolution of Distributed Ontology Process Models

An ontology is *replication consistent* if it is equivalent to its original and all its included ontologies (directly and indirectly) are replication consistent [8]. Fig.6 shows a distributed ontology system. *SO* and *CO* are defined at *A*; *BO* is defined at *B*, so *SO* must be replicated to *B*; *ICO* is defined at *C*, so *SO*, *BO* and *CO* must be replicated to *C*. *SO* at *B* is inconsistent if it has not been updated according to its original's changes at *A*; since *BO* at *B* includes *SO* which is inconsistent, then *BO* is inconsistent; so is *ICO* at *C*. Restrict that modifica-

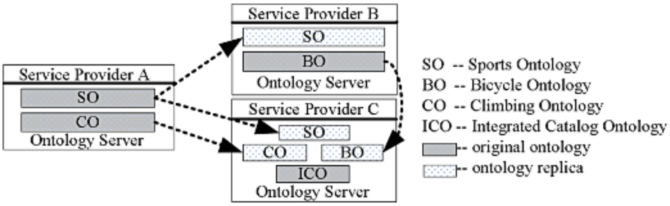


Fig. 6. Distributed ontologies

tion should always be directly performed at the original but not replicas and be propagated to the replicas, and adopt pull synchronization approach between originals and replicas. Replication consistency is performed by determining the equivalence of ontology with its original and by recursively determining the replication consistency of included ontologies [8]. For Fig.6, if C wants to resolve the replication inconsistency of ICO, then the pull synchronization process is realized as

$$(\overline{z}(BEGIN)|\overline{t}(BEGIN)).(z(msg_1)|t(msg_2)).[msg_1 = END][msg_2 = END]action_1.$$
$$ICO|z(msg_3).[msg_3 = BEGIN]\overline{c}(IFCONSISTENT).c(msg_4).([msg_4 = CONSIS$$
$$-TENT]\overline{x}(BEGIN).x(msg_5).([msg_5 = END]\overline{z}(END).action_2+[msg_5 = NOTREA$$
$$-DY]\overline{z}(NOTREADY)) + [msg_4 = INCONSISTENT]\overline{z}(NOTREADY)).BO|t(ms$$
$$-g_6).[msg_6 = BEGIN]\overline{b}(IFCONSISTENT).b(msg_7).([msg_7 = CONSISTENT]\overline{y}($$
$$BEGIN).y(msg_8).([msg_8 = END]\overline{t}(END).action_3 + [msg_8 = NOTREADY]\overline{t}(NOT$$
$$-READY)) + [msg_7{=}INCONSISTENT]\overline{t}(NOTREADY)).CO|(x(msg_9)|y(msg_{10})$$
$$).[msg_9 = BEGIN][msg_{10} = BEGIN]\overline{s}(IFCONSISTENT).s(msg_{11})([msg_{11} = CO$$
$$NSISTENT](\overline{x}(END)|\overline{y}(END)).action_4 + [msg_{11} = INCONSISTENT](\overline{x}(NOTR$$
$$-EADY)|\overline{y}(NOTREADY))).SO$$

The links between BO and SO, CO and SO, BO and ICO, CO and ICO are x, y, z and t respectively. b, c and s respectively denote the links of physical URI between the replicas of BO, CO and SO at C and their originals. *IFCONSISTENT*, *CONSISTENT* and *INCONSISTENT* are the messages used for the synchronization between originals and replicas. *NOTREADY* is returned by the included replica when its original is inconsistent and the process will be suspended. $action_1$ is caused by the replication inconsistencies of BO, CO and SO on ICO; $action_2$ and $action_3$ are respectively caused by itself and SO on BO and CO; $action_4$ is caused by itself on SO.

5 Related Work

Much related work has been done on the ontology evolution investigation. [11] proposes an approach for analyzing and classifying the operations on ontology. [12] discusses OntoView, a web-based change management system for ontologies. [8] presents an approach for evolution in the context of dependent and distributed ontologies. [13] presents an approach to model ontology evolution as the reconfiguration-design problem solving. A model transformation based conceptual framework for ontology evolution is presented in [14].

Pi-calculus is an expertise for describing mobile process and enables dynamic system modeling and synchronization detection. A large number of tools, such as JACK tool set and value-passing process algebra tool VPAM [15] are provided for correctness detection and related application. For the aptness for pi-calculus to model concurrent and dynamic systems, this paper proposes to manage the changes in ontology evolution using pi-calculus.

6 Conclusions and Further Work

Based on the pi-calculus, this paper proposes a kind of ontology process model, based on which, the change implementation and propagation problems of ontology evolution process are discussed at three levels: change implementation and precedence order of single ontology evolution, realization of the push-based synchronization for the change propagation in the evolution of multiple dependent ontologies within a single node, and the realization of the pull synchronization for the change propagation of distributed ontologies evolution.

A lot of work is needed to do, just explain a few. To refine the ontology process model and the elementary change operations, to define the complex change operations, to consider the change validation based on the model and develop the tools supporting the evolution process for single ontology, multiple dependent ontologies within a single node and distributed ontologies.

References

1. D. Fensel. Ontologies: dynamics networks of meaning. In Proceedings of the 1st Semantic web working symposium, Stanford, CA, USA, 2001.
2. M. Klein, and D. Fensel. Ontology versioning for the Semantic Web. In Proceedings of International Semantic Web Working Symposium, USA, 2001.
3. Ljlijana Stojanovic. Methods and Tools for Ontology Evolution. PhD thesis, University of Karlsruhe, 2004.
4. Ljiljana Stojanovic, Alexander Mädche, Boris Motik, and Nenad Stojanovic. User-driven ontology evolution management. In Proceedings of the 13th European Conference on Knowledge Engineering and Management (EKAW 2002), number 2473 in Lecture Notes in Computer Science, pages 285-300, Siguenza, Spain, October 2002. Springer-Verlag.
5. U. Nestmann, and B. Victor. Calculi for mobile processes: Bibliography and web pages. Bulletin of the EATCS, 64: 139-144, 1998.
6. J. A. Bergstra, A. Ponse, and S. A. Smolka, editors. Handbook of Process Algebra. Elsevier, 2001.
7. Peter Haase, and Ljiljana Stojanovic. Consistent Evolution of OWL Ontologies. In Proceedings of the 2nd European Semantic Web Conference (ESWC 2005), number 3532 in Lecture Notes in Computer Science, pages 182-197, Heraklion, Greece, May 29-June 1, 2005. Springer-Verlag.
8. Alexander M., Boris M., and Ljiljana S. Managing multiple and distributed ontologies in the semantic web. VLDB Journal, 12(4): 286-302, 2003.

9. Bhide M., Deoasee P., Katkar A., Panchbudhe A., and Ramamritham K. Adaptive push-pull: disseminating dynamic Web data. IEEE Trans Comput, 51(6): 652-668, 2002.
10. Pierre G., and van Steen M. Dynamically selecting optimal distributing strategies on Web documents. IEEE Trans Comput, 51(6): 637-651, 2002.
11. Paolo Ceravolo, Angelo Corallo, Gianluca Elia, and Antonio Zilli. Managing Ontology Evolution Via Relational Constraints. In Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES 2004), number 3215 in Lecture Notes in Artificial Intelligence, pages 335-341,Wellington, New Zealand,September 2004. Springer-Verlag.
12. Michel K., Atanas K., Damyan O., and Dieter F. Finding and characterizing changes in ontologies. In Proceedings of the 21st International Conference on Conceptual Modeling(ER2002), number 2503 in Lecture Notes in Computer Science, pages 79-89, Tampere, Finland, October 2002. Springer-Verlag.
13. Ljiljana Stojanovic, Alexander Maedche, Nenad Stojanovic, and Rudi Studer. Ontology evolution as reconfiguration-design problem solving. In Proceedings of the 2nd International Conference on Knowledge Capture (KCAP 2003), pages 162-171, Sanibel, Florida, October 2003. ACM, OCT.
14. Longfei Jin, Lei Liu, and Dong Yang. A Model Transformation Based Conceptual Framework for Ontology Evolution. In Proceedings of the 9th International Conference on Knowledge-Based, Intelligent, Information, and Engineering Systems (KES 2005), number 3681 in Lecture Notes in Artificial Intelligence, pages325-331, Melbourne, Australia, September 2005. Springer-Verlag.
15. Huimin Lin. A verification tool for value-passing process algebras. IFIP Transactions C-16: Protocol Specification, Testing and Verification, North-Holland, 1993, 79-92.

A Comprehensive Study of Inappropriate Hierarchy in WordNet

Liu Yang

Institute of Computatinal Linguistics, Peking University
Bejing 100871, China
liuyang@pku.edu.cn

Abstract. In WordNet, the lexicalized noun/ verb concepts are organized hierarchically by means of hypernymy. As the most basic semantic relation, hypernymy not only serves to construct the specific hierarchy of the concepts in the domain, but also provides a common way of reasoning along the hierarchy for NLP researchers. However, we've found two kinds of inappropriate hierarchy in WordNet, the cases of ring and isolator for short. This paper offers a comprehensive study of these cases and make clear that they can cause a degenerate structure, hence harass the reasoning and eventually lead to errors.

1 Introduction

WordNet is a lexical database in which semantics of English content words, such as nouns, verbs, adjectives and adverbs, are described and presented by a certain approach. The approach first emerged from G. A. Miller's an inspiration at 1984 and then an actual lexical project proceeded at Princeton University. After 20 years of R&D contributed by G. A. Miller, C. Fellbaum and their colleagues, WordNet nowadays has become one of the most widely used language ontologies in the field of semantic analysis for NLP researchers around the world [6][7].

In contrast to the constructive approach to the presentation of lexical semantics, the approach adopted by WordNet is a differential one [8]. WordNet first defines the lexicalized concept by a less abstract form, synset, namely synonym set. Further, the synsets are linked by means of a number of semantic relations, including hypernymy/ hyponymy, holonymy/ meronymy, cause, entailment, and so on. All these synsets, as the nodes, together with all these semantic relations, as the arcs, have interlaced into a huge network. In such a network, a word sense is entirely determined by the specific position of a corresponding synset in which the word may occur among others. NLP researchers can benefit a lot by exploiting the network without considering such troublesome things as lexemes or sememes.

Among all the semantic relations, hypernymy is by all means the most basic one, which serves to organize the noun/ verb concepts hierarchically. Only after its role established as the key organizer of the backbone structure of the network, can other semantic relations be added into the hierarchies [6]. This is the organization of the noun/ verb concepts in WordNet. With respect to that of the adjective/ adverb concepts, things are quite different. Nothing like hypernymy that generates nominal hierarchies

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 639 – 645, 2006.
© Springer-Verlag Berlin Heidelberg 2006

can be available for the modifiers. It is antonymy that acts as the basic semantic relation to organize the adjective/ adverb concepts into shallow clusters or just separate lines [3].

This particular organization of the noun/ verb concepts then provides a common way of reasoning in applications. The NLP researchers can, somehow, locate and evaluate the sense of a word along the hierarchy. The reasoning of a given ontology, say induction and deduction in WordNet, thus gets involved. From this viewpoint, the hierarchies should always be kept well formed in WordNet. This is just a natural assumption. However, we've found two kinds of inappropriate hierarchy in WordNet, the cases of ring and isolator for short [5], which can cause a degenerate structure, hence harass the reasoning in NLP practice and eventually lead to errors.

This paper tries to offer a comprehensive study of these two cases of inappropriate hierarchy. In section 2, we make an introduction to the hierarchy theory of WordNet and its taxonomies to set a criterion on which we can judge something later. By this criterion, Section 3, form a more theoretical viewpoint, elaborates on how and why the inappropriate hierarchy may occur. In section 4 and 5, the most dramatic cases of ring and isolator are demonstrated respectively, together with their distribution in different domains. And at the end is the conclusion.

2 The Hierarchy Theory of WordNet and Its Taxonomies

As a useful and efficient means of defining the nominal things, information about hypernymy between nouns is often given in conventional dictionaries. As a simplified example, the sense of a bird *robin* might be defined by a phrase something like a *migratory bird* that has a clear melodious song and a reddish breast with gray or black upper plumage. This shows a common definitional formula [1][3]. It consists of a hypernym or genus term, preceded by adjectives or followed by relative clauses that describe how this instance differs from all other instances of that hypernym. Since the purpose of a lexical definition is to distinguish among hyponyms and much information can be inherited in this manner, there is no need to enumerate all features of the word's referent separately.

WordNet adopts this way of definition in conventional dictionaries and extends it. In WordNet, hypernymy is actually depicted as a relation between the lexicalized concepts, represented by a pointer @ between the appropriate synsets. At the same time, the researchers at Princeton University noticed that, in conventional dictionaries, loops sometimes do arise inadvertently as the dictionaries are oriented to human. For example, word W_a is used to define word W_b whereas word W_b is used to define word W_a.

As for a MRD, the decision of the researching team is that circularity is the exception, not the rule, and should always be avoided in WordNet. So the fundamental design that lexicographers try to impose on the semantic organization of nouns is not a circle, but a hierarchy [3]. This is a settlement of issue with the upward linkage of hypernymy as a semantic relation. When comes to the downward side of hypernymy, say how a hyponym differs from others, the notion of the distinguishing features is further introduced by WordNet. They are attributes, parts and functions. These features are centrally important as hyponym is actually defined by them [3].

Thus, a lexical hierarchy can be reconstructed by following the trail of hypernymy links, {*robin, redbreast*} @-> {*bird*} @-> {*animal, animate_being*} @-> {*organism, life_form, living_thing*} for example. This creates a sequence of levels, going from many specific terms at the lower levels to a few generic terms at the top. By combining the potential hierarchies in different domains, the nouns in WordNet form a lexical inheritance system [3].

As far as the verb concepts are concerned, the same organization is constructed in WordNet, with more or less modification of the definition of hypernymy between the verb concepts. Such is the hierarchy theory of WordNet.

Concerning the taxonomies of whole set of the concepts, WordNet divides the noun concepts into 25 hierarchies, each with a different unique beginner. This multiple hierarchies correspond to relatively distinct semantic fields or domains. Since the features that characterize a unique beginner are inherited by all of its hyponyms, a unique beginner corresponds roughly to a primitive semantic component in a compositional theory of lexical semantics [3]. Partitioning the noun concepts also has a practical advantage of reducing the size of the files and assigning them to different lexicographers. These hierarchies vary widely in size and are not mutually exclusive. Some cross-referencing is required, but, on the whole, they cover distinct conceptual domains. These independently exist 25 taxonomic trees, in lines with 25 hierarchies, for all these domains although some tree nodes between different trees may interlace via hypernymy occasionally [3].

Among the forest of the first 25 trees, WordNet also makes a higher level of some groupings, with 8 concerned domains grouped as {*entity*}, 5 domains as {*abstract*}, and another 3 domains as {*psychological_feature*}. Thus this forest is reduced to 11 trees. To include information about the higher level, another additional domain, named Noun Top, with the domain ID of 3, is added into the system. The names of the domains of the noun concepts, together with their domain ranging from 3 to 28, are listed omitted here.

As for the taxonomies of whole set of the verb concepts, things of the 15 hierarchies or taxonomic trees are quite similar to those of the noun concepts. These also independently exist 15 taxonomic trees, in lines with 15 hierarchies although they tend to be shallower and bushy compared with those of the noun concepts.

Some nodes between different trees may also interlace via hypernymy occasionally [3][4]. But there is no a higher level of some groupings among the forest of the first 15 trees any more. The names of the domains of the verb concepts, together with their domain ID ranging from 29 to 43, are also omitted.

3 The Inappropriate Hierarchy May Occur in WordNet: How and Why

In principle, hypernymy, as a semantic ralation, indicates the uniqueness of induction by its original meaning and the hypernym of a word should always be in the same semantic domain of the word proper. This is quite true of the general linguistics theory. We, however, live in a world of reality other than theory. There do exist case that it is hard to reach the uniqueness of induction for a certain lexicalized concept and we can only hold such a belief that this concept might have more than one hypernym, one in its own domain, the main domain and the others in other domains, the less prominent domains. This is an exception.

Such is the background of the strategies WordNet adopted. As we mentioned above, these do independently exist taxonomic trees, in lines with hierarchies, for all noun/ verb domains although some tree nodes between different trees may interlace by hypernymy occasionally. This is the phenomenon of multiple inheritance or multiple parentage in WordNet. In other words, there exist such circles between different taxonomic trees and the taxonomies, constructed by hypernymy, of the noun/ verb concepts in WordNet are DAGs rather than TREEs in sense of their topology. This fact has been noticed by a few of researchers [2][3]. But, until now, no person has recognized the different conditions of such circles and there are also no analyses of such issues as whether or not the circles are between different taxonomic trees or just in an independent taxonomic tree. They are rather different things concerning their different acceptability in construction and reasoning in practice.

Also, in contrast to the phenomenon of multiple inheritance or multiple parentage in WordNet, the phenomenon of some kind of zero inheritance or zero parentage, other than that of the root of the taxonomic tree, has not been noticed yet. Nobody wakes up to the fact that there may exist some special orphans in WordNet.

Hence we carry a systematic and theoretical study of these phenomena, which, on the whole, have something to do with the hierarchy theory or inheritance system adopted by WordNet.

If we use H_{in} to measure the hypernyms of a certain concept C_x in its own semantic domain and H_{out} to measure its hypernyms in other domains, the cases we can accept should always satisfy the condition of $0<H_{in}<=1$ and the value of H_{out} does not matter too much since our goal is to get a taxonomy in a certain domain, with a certain set of criteria, in order to get a certain taxonomic tree.

Then what happens to the cases not satisfying this condition? What is the meaning of these exceptional cases and whether or not these shall happen in WordNet 2.0, the latest version of WordNet family by now?

The denial of $0<H_{in}<=1$ might yield either $H_{in}>=2$, case 1 for short, or $H_{in}=0$, case 2 for short. As the root of the taxonomic tree can also satisfy condition of case 2, we strengthen the condition of case 2 by adding $H_{out}>=1$ to it and then get condition of $H_{in}=0$ and $H_{out}>=1$, case 3 for short.

For case 1, condition of $H_{in}>=2$ means that the current concept C_x has at least 2 hypernyms in its own semantic domain. According to the specification of WordNet we've mentioned above, each domain already denotes a taxonomic tree by hypernymy. This condition will unavoidably lead to the case of ring, in terms of an undirected graph derived from the corresponding DAG, in WordNet. Along these upward arcs of hypernymy of concept C_x, there naturally exists C_x's most nearby ancestor, say concept C_z, which has at least 2 hyponyms, say concept C_{y1} and C_{y2}; at the same time, concept C_{y1} and C_{y2} are all C_x's ancestors. As WordNet is an inheritance system, we can now infer that C_x shares C_{y1} and C_{y2}'s all features or properties, among which a pair of features or properties must be different for C_{y1} and C_{y2} have the same hypernym C_z and hereby is distinguishable. This is rather paradoxical by the general linguistic theory. This is nonsense and leads to the case of ring. For case 2, condition of $H_{in}=0$ means that, as we already mentioned earlier, the current concept C_x has no hypernym at all and it might be the root of the taxonomic tree. This condition doesn't lead to any faults. For case 3, condition of H_{in}=0 and H_{out}>=1 means that the current concept C_x has no any available concept C_z as its hypernym in its own domain. Rather, C_x has at least 1 hypernym in other domains and actually belongs to those domains. This is nonsense and leads to the case of isolator.

In the final analysis, both the cases of ring and isolator, if they occur in WordNet, are abnormal and unacceptable with respect to the specification.

4 Inappropriate Cases of Ring in WordNet

In order to explore the actual cases of ring and the amount of them in WordNet, we devised the searching algorithm for cases of ring for the noun concepts. It can also apply to the verb concepts easily after the minor modifications of the value of the boundary information about the semantic domains. By this powerful searching algorithm, we found 1,839 occurrences out of a total of 79,689 noun concepts and 17 occurrences out of a total of 13,508 verb concepts for the case of ring in WordNet 2.0. The percentages are 2.31% and 0.13% respectively. Table 1 and 2 show the detailed portion for each domain.

Table 1. Occurrences of cases of ring in each semantic domain for the noun concepts

[ID=04]	73	[ID=09]	29	[ID=14]	34	[ID=19]	7	[ID=24]	2
[ID=05]	27	[ID=10]	67	[ID=15]	205	[ID=20]	29	[ID=25]	4
[ID=06]	258	[ID=11]	5	[ID=16]	0	[ID=21]	10	[ID=26]	102
[ID=07]	12	[ID=12]	11	[ID=17]	11	[ID=22]	8	[ID=27]	193
[ID=08]	23	[ID=13]	24	[ID=18]	682	[ID=23]	13	[ID=28]	8

Table 2. Occurrences of cases of ring in each semantic domain for the verb concepts

[ID=29]	0	[ID=32]	0	[ID=35]	4	[ID=38]	1	[ID=41]	2
[ID=30]	5	[ID=33]	0	[ID=36]	2	[ID=39]	0	[ID=42]	1
[ID=31]	0	[ID=34]	1	[ID=37]	0	[ID=40]	1	[ID=43]	0

Figure 1 is a demo of the case of ring, with Min_Length=2 and domain ID=3, for the noun concepts. The unit in the form (03, 00001740, {*entity*}) represents domain ID, offset and synset respectively. And the topmost node of the directed graph is a root node of a certain taxonomic tree in WordNet. The arc in the graph represents hypernymy. It's fairly astonishing that, in such high level of the hierarchy and confined to such minimus length of a loop, the case of ring does exist. This kind of things is abnormal and appropriate considering the hierarchy theory of WordNet and the criterion for taxonomy as mentioned in section 2 as well as our analyses in section3.

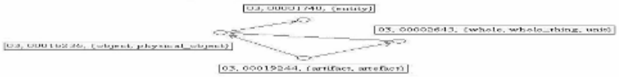

03, 00001740, {entity}
03, 00002645, {whole, whole_thing, unit}
03, 00016236, {object, physical_object}
03, 00019244, {artifact, artefact}

Fig. 1. A case of ring with Min_Length=2 in WN2.0 noun concepts

Figure 2 is a demo of the case of ring, with Min_Length=2 and domain ID=30, for the verb concepts. Things are quite similar to the above mentioned case for the noun concepts. But, more directly in this case, {*turn*} must inherit the distinguishing features or attributes from both {*turn, grow*} and {*discolor, discolour, colour, color*} at the same time as they share the identical hyernym {*change*}. This is a dilemma.

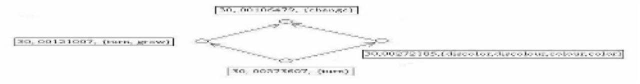

30, 00126477, {change}
30, 00131007, {turn, grow}
30,00272105,{discolor,discolour,colour,color}
30, 00273607, {turn}

Fig. 2. A case of ring with Min_Length=2 in WN2.0 verb concepts

5 Inappropriate Cases of Isolator in WordNet

The searching algorithm for cases of isolator is rather simple in contrast to noticing this phenomenon proper. We don't elaborate on it here any more. For the case of ring, there are 2,654 occurrences out of a total of 79,689 noun concepts and 1,551 occurrences out of a total of 13,508 verb concepts in WordNet 2.0. The percentages are 3.33% and 11.48% respectively. Table 3 and 4 show the details in domains.

Table 3. Occurrences of cases of isolator in each semantic domain for the noun concepts

[ID=04]	65	[ID=09]	54	[ID=14]	37	[ID=19]	33	[ID=22]	10
[ID=05]	415	[ID=10]	73	[ID=15]	351	[ID=20]	286	[ID=23]	15
[ID=06]	199	[ID=11]	15	[ID=16]	6	[ID=21]	56	[ID=24]	72
[ID=07]	30	[ID=12]	42	[ID=17]	114	[ID=22]	10	[ID=25]	21
[ID=08]	93	[ID=13]	34	[ID=18]	394	[ID=23]	15	[ID=28]	28

Table 4. Occurrences of cases of isolator in each semantic domain for the noun concepts

[ID=29]	104	[ID=32]	136	[ID=35]	283	[ID=38]	106	[ID=41]	197
[ID=30]	211	[ID=33]	69	[ID=36]	43	[ID=39]	45	[ID=42]	112
[ID=31]	87	[ID=34]	32	[ID=37]	36	[ID=40]	76	[ID=43]	14

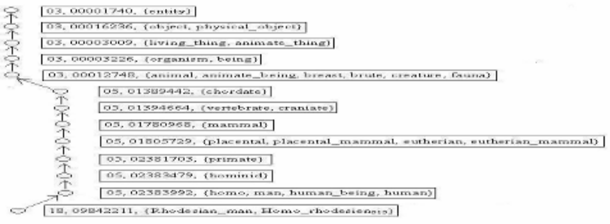

03, 00001740, {entity}
03, 00016236, {object, physical_object}
03, 00003009, {living_thing, animate_thing}
03, 00003226, {organism, being}
03, 00012748, {animal, animate_being, breast, brute, creature, fauna}
05, 01389442, {chordate}
05, 01394664, {vertebrate, craniate}
05, 01780968, {mammal}
05, 01805729, {placental, placental_mammal, eutherian, eutherian_mammal}
05, 02381703, {primate}
05, 02383479, {hominid}
05, 02383992, {homo, man, human_being, human}
18, 09842211, {Rhodesian_man, Homo_rhodesiensis}

Fig. 3. A case of isolator with Max_Length=12 in WN2.0 noun concepts

Figure 3 is a demo of the case of isolator, with Max_Length=12 and domain ID=18, for the noun concepts. It's also amazing that, after inheriting so many specific features and attributes along the linkage of hypernymy, the bottom most concept actually has nothing to do with its own domain at all. As a lost orphan of its own family as already depicted in the database, it immediately and ultimately falls into other domains going up a very long journey.

Figure 4 is a demo of the case of isolator, with Max_Length=11 and domain ID=41, for the verb concepts. A very long journey is also made meaninglessly.

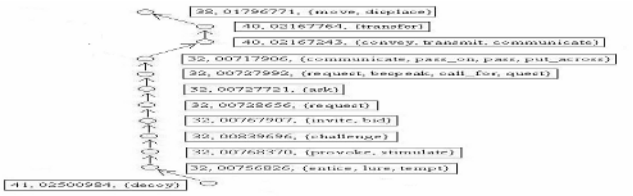

32, 01796771, (move, displace)
40, 03167764, (transfer)
40, 02167243, (convey, transmit, communicate)
32, 00717906, (communicate, pass_on, pass, put_across)
32, 00727992, (request, bespeak, call_for, quest)
32, 00727721, (ask)
32, 00728656, (request)
32, 00767907, (invite, bid)
32, 00829696, (challenge)
32, 00768370, (provoke, stimulate)
32, 00756826, (entice, lure, tempt)
41, 02500924, (decoy)

Fig. 4. A case of isolator with Max_Length=11 in WN2.0 verb concepts

6 Conclusion

To sum up, the cases of ring and isolator, as two kinds of inappropriate hierarchy, can cause a degenerate structure of WordNet, hence harass the reasoning in NLP practice and eventually lead to errors.

In the future, more exploration and amendment should be made to solve these issues during the evolution of WordNet family.

Acknowledgments. This research is supported by the National Hi-tech R&D Program of China and the FANEDD Project of Chinese Ministry of Education, under grant No. 2004CB318102, 200514 respectively.

References

1. Touretzky, D. The Mathematics of Inheritance System. Los Altos, CA: Morgan Kaufman (1986).
2. Devitt, A. and Vogel, C. The Topology of WordNet: Some Metics. In Proc. of GWC'04, pp106-111, Czech (2004).
3. Fellbaum, C. (eds.), WordNet: an Electronic Lexical Database. Cambridge, Mass.: MIT Press (1998).
4. Fellbaum, C. English Verbs as a Semantic Net. Technical Report of Cognitive Science Laboratory, Princeton Univ. (1993).
5. Liu, Y., Yu, S. W. and Yu, J. S. Two Kinds of Hypernymy Faults in WordNet: the Cases of Ring and Isolator. In Proc. of GWC'04, pp347-351, Czech (2004).
6. Liu, Y., Yu, S. W. and Yu, J. S. Building a Bilingual WordNet-Like Lexicon: the New Approach and Algorithms. In Proc. of COLING'02, pp1243-1247, Taipei, China (2002).
7. Vossen, P. (eds.), EuroWordNet: a Multilingual Database with Lexical Semantic Networks. Dordrecht: Kluwer (1998).
8. Wong, P. W. and Fung, P. Nouns in Wordnet and HowNet: An Analysis and Comparison of Semantic Relations. In Proc. of GWC'02, pp319-322, India (2002).

Autonomous Ontology: Operations and Semantics OR Local Semantics with Semantic Binding on Foreign Entity

Yuting Zhao[1], Luciano Serafini[1], and Fausto Giunchiglia[2]

[1] ITC-IRST
Via Sommarive 18
38050 Povo, Trento, Italy
{yzhao, luciano.serafini}@itc.it
[2] Department of Information and Communication Technology
University of Trento
Via Sommarive 14
38050 Povo, Trento, Italy
fausto@dit.unitn.it

Abstract. In this paper, firstly we put forward an *AO framework* of autonomous ontology, by which a language entity in an ontology is interpreted locally, nevertheless the semantic cooperation is still keeping. Different from various of works based on DDL (Distribute Description Logic [1]), *AO* framework depends on *semantic binding* instead of domain-relation to set up relationship between different ontologies so that it avoids damaging the autonomy of each local ontology. Secondly we formalize the basic operations among ontologies, say *free-access operation*, *importing operation*, and *mapping operation* in the *AO* framework, and give out the proper semantics of them.

1 Introduction

In order to develop a semantical WEB environment such that computer-based agent is able to "understand" the meaning of the data it is accessing, a framework of knowledge representation which supports semantical cooperations between distributed knowledge-base is needed. Ontology is a formalization of the structure of a conceptualized domain, which is suitable to present common opinion on a domain from different parties. Ontology space [2], which is a set of ontologies, is introduced to fulfill the "open" and "distributed" requirement of semantic Web. Accordingly relationship and coordination among ontologies have became focused topic in the field of Semantic WEB.

In this paper, firstly we put forward an *AO framework* which supports a distributed autonomous ontology space. In *AO* framework, on the one hand a language entity in an ontology is interpreted locally in order to keep the autonomy, on the other hand a language entity is constrained by a *semantic binding* if necessary, in order to enable semantic cooperation. Different from various of

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 646–652, 2006.
© Springer-Verlag Berlin Heidelberg 2006

works (like [2], [3], [4], and [5]) based on DDL (Distribute Description Logic [1]), AO framework depends on semantic binding in stead of domain-relation to build relationship between different ontologies, so that it avoids damaging the autonomy of each local ontology.

In the second part of this paper, we discuss on three basic operations among ontologies, say *free-access operation*, *importing operation*, and *mapping operation*, and then formalize them in the AO framework.

In order to respects the "open world" principle of WEB, The Web Ontology Language (OWL) [6] allows that one can freely use the URI [7] to build up ontology and express herself. The free-access operation sets up the cooperation between ontologies; one ontology is saying something about another. By free-access one ontology can 'cite' the class, property, and individual of the other ontologies in its own TBox or ABox. Nevertheless, following the "open world" principle of WEB, the free-access operation does litter on the semantics cooperation.

The *owl:imports* operation allows to include all the axioms in the imported ontology into another ontology. Obviously by this operation one imports the whole ontology, so that the semantical information is transferred. We note that the importing operation is transitive, so we introduce a *full ontology* to unfold an ontology having *owl:imports*, and give a proper semantics on it.

Semantic mappings are used to set up partially knowledge sharing among ontologies. We note that, the approach given by DDL [1] actually realizes the transferring of semantical information by domain relations. Semantics of the bridge rule of one ontology depends on the domain of the other ontology as well as how the latter ontology is interpreted. It damages the autonomy of each local ontology, and forces the user to treat all of the ontologies as a whole. In this paper we use the *contextual mapping* by [2] as example, and illuminate how to transfer a domain-relation based mapping into AO framework.

2 Ontology Space and Foreign Entities

In general understanding, ontology in the field of semantic web is a set of sharing knowledge instead of an unique description of the universe. Also because modern WEB is of "distributed", the idea of multiple ontologies is accepted by the society.

Definition 1 (*Ontology Space*). *Let I be a set of indexes, standing for a set of URIs for ontologies. Let $\mathbb{L}_I = \{\mathbb{L}_i\}_{i \in I}$ be a set of languages. An* Ontology Space *OS on $\mathbb{L}_I$ is a family $\{O_i\}_{i \in I}$,* s.t. *every O_i is an ontology on language $\mathbb{L}_i$, where $i \in I$.*

In ontology space $OS = \{O_i\}_{i \in I}$, we denote, by $\mathbb{C}_i$ the set of concept names in ontology O_i. Analogous to $\mathbb{R}_i$ and $\mathbb{O}_i$. Actually *language* $\mathbb{L}_i$ is the disjoint union of $\mathbb{C}_i$, $\mathbb{R}_i$ and $\mathbb{O}_i$.

In ontology space, sometimes a language entity (concept, role, or individual) is defined in one ontology, but could be used in another ontology. So we partition the language $\mathbb{L}_i$ in two parts: the *local entity* and the *foreign entity* (named local language and foreign language in [2]). Intuitively, local entities are the roles, concepts, and individuals that one invites in her own ontology; foreign entities are the roles, concepts, and individuals that she borrows from the other ontologies in order to define something in her ontology.

In this paper, when we are talking about semantics and reasoning, we always tell a language entity in the ontology space by a way showing (1) where it is using, and (2) where it is originally defined. Suppose that $C \in \mathbb{L}_i$ and $i, j \in I$, then formally in ontology space we have a language entity like,

$$i\!:\!j\!:\!C \tag{1}$$

which means a language entity C is appeared in ontology O_i, but originally defined in ontology O_j. This kind of denotation is applied to concepts, roles, and individuals in ontology space. At the same time, we still use the namespace-like notation in [2] in the description of syntax (also abstract syntax) in this paper.

3 Autonomous Ontology

In one ontology space, each ontology reflects the subjective opinion on a partial structure of the universe. In semantic web, in general one presents her personal knowledge (understanding) by her ontology. Thereafter we argue each ontology should be semantical independent and keeping autonomy.

On the one hand, we apply *local interpretation* to local entity as well as foreign entity in an ontology, in order to keep the semantical autonomy; on the other hand, we introduce *semantic bindings* on foreign entities, in order to realize semantic cooperation among an ontology space.

For an ontology space $OS = \{O_i\}_{i \in I}$, $\mathcal{I} = \{\mathcal{I}_i\}_{i \in I}$ is a *local interpretation*, iff every ontology O_i has an interpretation $\mathcal{I}_i = \left\langle \Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}_i} \right\rangle$, in where $\Delta^{\mathcal{I}_i}$ is the local domain of O_i, and $\cdot^{\mathcal{I}_i}$ is a mapping that assigns to each concept name C a subset of $\Delta^{\mathcal{I}_i}$, to each role name R a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_i}$, and to each individual name o an element of $\Delta^{\mathcal{I}_i}$.

Following common understanding, for an ontology O_i if an axiom a is true under an interpretation $\mathcal{I}_i$, we say that the axiom a is *satisfied* by $\mathcal{I}_i$, and denote this by $\mathcal{I}_i \models a$. We say $\mathcal{I}_i$ is a *model* of an ontology O_i, if $\mathcal{I}_i$ satisfies all axioms in O_i. If formula ϕ is true under all of the models of an ontology O_i, we say that O_i entails ϕ, and denote this by $O_i \models \phi$.

Now we introduce *semantic binding* between a foreign entity and itself in its original ontology, so that the semantics of a language entity in its original ontology is transferred into the current ontology.

For $i \neq j \in I$, a *semantic binding* from O_i to O_j on foreign entity $j\!:\!i\!:\!x$ is an expression of $i\!:\!i\!:\!x \overset{\equiv}{\Longrightarrow} j\!:\!i\!:\!x$, which shows foreign entity $j\!:\!i\!:\!x$ in ontology O_j is bound to the semantics of local entity $i\!:\!i\!:\!x$ in ontology O_i. In this paper use $\mathbb{B}_j$to denote the set of foreign entities under semantic binging in ontology O_j.

Definition 2 (Consistency of Semantic Binding). *Let $\mathcal{I}_j$ a model of ontology O_j. We say $\mathcal{I}_j$ is* consistent *to the semantic bingings $\{b\ |b \in \mathbb{B}_j\}$ in ontology space $OS = \{O_i\}_{i \in I}$, iff, for any $j\!:\!i\!:\!\mathbf{x}$, $j\!:\!i\!:\!\mathbf{y}$, $j\!:\!i\!:\!\mathbf{z}$ $\in \mathbb{B}_j$, we have*

1. *if* $j{:}i{:}\mathbf{x},\ j{:}i{:}\mathbf{y} \in \mathbb{C}_j$ *and* $O_i \models i{:}i{:}\mathbf{y} \sqsubseteq i{:}i{:}\mathbf{x}$*, then* $\mathcal{I}_j \models j{:}i{:}\mathbf{y} \sqsubseteq j{:}i{:}\mathbf{x}$*;*
2. *if* $j{:}i{:}\mathbf{x},\ j{:}i{:}\mathbf{y} \in \mathbb{R}_j$ *and* $O_i \models i{:}i{:}\mathbf{y} \sqsubseteq i{:}i{:}\mathbf{x}$*, then* $\mathcal{I}_j \models j{:}i{:}\mathbf{y} \sqsubseteq j{:}i{:}\mathbf{x}$*;*
3. *if* $j{:}i{:}\mathbf{x},\ j{:}i{:}\mathbf{y} \in \mathbb{O}_j$ *and* $O_i \models i{:}i{:}\mathbf{y} \equiv i{:}i{:}\mathbf{x}$*, then* $\mathcal{I}_j \models j{:}i{:}\mathbf{y} \equiv j{:}i{:}\mathbf{x}$*;*
4. *if* $j{:}i{:}\mathbf{x} \in \mathbb{O}_j$*,* $j{:}i{:}\mathbf{y} \in \mathbb{C}_j$ *and* $O_i \models i{:}i{:}\mathbf{x} \in i{:}i{:}\mathbf{y}$*, then* $\mathcal{I}_j \models j{:}i{:}\mathbf{x} \in j{:}i{:}\mathbf{y}$*;*
5. *if* $j : i : \mathbf{x},\ j : i : \mathbf{y}\ \in \mathbb{O}_j$*,* $j : i : \mathbf{z}\ \in \mathbb{R}_j$ *and* $O_i \models (i : i : \mathbf{x}, i : i : \mathbf{y}) \in i : i : \mathbf{z}$*, then* $\mathcal{I}_j \models (j{:}i{:}\mathbf{x}, j{:}i{:}\mathbf{y}) \in j{:}i{:}\mathbf{z}$*.*

We denote this fact by $OS \models_{AO} \mathcal{I}_j$*, and call* $\mathcal{I}_j$ *an AO model of* OS*.*

Semantic binding guarantees that a foreign entity keeps the semantics in the ontology it is original defined. For example, if in the original ontology, x is a subsumption of y, then as foreign concepts in another ontology which is consistent with corresponding semantics bindings, they still remain this "subsumption" relationship.

An *AO framework* is a pair $\langle OS, B\rangle$ in where $OS = \{O_i\}_{i\in I}$ is an ontology space and B is a set of semantic binding on OS, $B = \{B_i\ |B_i$ *is the semantic binging of* $O_i \in OS\}$. In next section we show how *AO* framework is suitable to describe operations and relationships between ontologies.

4 Operations in Ontology Space

In an ontology space, relationships between ontologies are realized by operations. In this paper we discuss three basic operations between ontologies: *free-access operation*, *importing operation*, and *mapping operation*. Intuitively, free-access operation enable the ontology space opening to anybody; importing operation allows some ontology to copy other ontology; mapping operation sets up semantical mapping between ontologies. In this section, we declare the semantics of these operations in *AO* framework.

4.1 Free-Access Operation

In the Web, people respects and follows an "open world" principle, that is "any body could say anything about anybody (at any time)". Furthermore in OWL, one can freely use the URI [7] to build up ontology and express herself. When we come to the ontology space, by which we pay more attention to the relationship and collaboration among ontologies, we note that the *free-access*, by the mean of free using of URI, is one of the basic operations in *AO* framework.

This operation sets up the cooperation between ontologies; one ontology is saying something about another. By free-access one ontology can 'cite' the class, property, and individual of the other ontologies in its own TBox or ABox.

Now we formalize the free-access relationship by autonomous ontology.

Definition 3 (Free-access Entity). *Let* α *be a foreign entity in ontology* O_i*. We say* α *is a free-access entity if it is not under semantic binding in an* AO *framework, i.e.* $\alpha \notin \mathbb{B}_i$*.*

As shown in above Section-3, we note that foreign entity without semantics binding has local semantics; it is interpreted in local domain, and there is no relation with its original definition. So the semantics of free-access entity in autonomous ontology reflects the spirit of the "open world" principle of WEB.

4.2 Importing Operation

Importing operation is also provided by the OWL language; by a built-in ontology properties *owl:imports*, the axioms in the imported ontology can be used in the current ontology [6].

In this paper, we use *importing*(O_i, O_j) to denote the relation that ontology O_i imports ontology O_j. We call O_i the *current ontology*, and O_j the *imported ontology* for the importing.

Since *Importing* is transitive, we introduce *importing closure* to present a set which contains the ontology and all ontologies (also their importing closure) being imported by it, and then unfold the importing transition by *full ontology.*

Definition 4 (Importing Closure). *For an ontology O, the* importing closure *is a set of ontologies, denoted by O^{IC}, such that,*

1. *$O \in O^{IC}$;*
2. *if $O_i \in O^{IC}$, and importing(O_i, O_j), then $O_j \in O^{IC}$.*

Definition 5 (Full Ontology). *For an ontology O, its* full ontology, *denote by O^{full} is constructed as followings:*

1. *Let $O^{full} = O$;*
2. *erases all* imports *annotations from O^{full};*
3. *let $O' = O^{IC} - \{O\}$, for any $O_i \in O'$, duplicates O_i in O^{full}.*

Definition 6 (Imported Foreign Entity). *For an ontology O_i in AO framework, we say a foreign entity $i : j : \beta$ is an imported foreign entity if there is importing(O_i, O_j).*

We note that an imported foreign entity is also a free-access entity, since there is not semantic binding. But the semantic of this kind of entity is still transferred from the original ontology because of the importing.

We note the interpretation of an ontology O with importing operations is exactly an interpretation of its full ontology O^{full}.

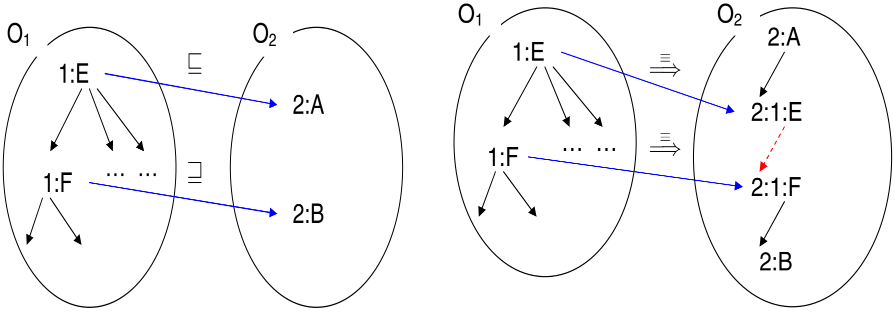
Definition 7 (Semantics of Importing). *For an ontology O with imports annotations , a set of atoms M is a model of O in AO framework iff it is a model of O^{full}.*

4.3 Semantic Mapping

Semantic mappings are used in approaches to partially share knowledge among ontologies. In this Section we use the *contextual mapping* by [2] as example to show how it is formalized in *AO* framework. This frame is able to extended to other kind of mappings.

A contextual mapping is a set of *bridge rules.* A bridge rule from O_i to O_j, like $i{:}i{:}x \xrightarrow{\sqsubseteq} j{:}j{:}y$, is intend to transfer semantical information from ontology O_i to ontology O_j. The meaning of this one is, from the point of view of ontology

Table 1. Example: Formalize mappings in AO framework



(a) Mappings

(b) Formalized in AO framework

O_j, an entity $i:i:x$ defined in O_i is less general than entity $j:j:y$, which is defined by itself. Semantics of the bridge rule of one ontology depends on the domain of the other ontology as well as how the latter ontology is interpreted. It damages the autonomy of each ontology, and forces the user to treat all of the ontologies as a whole. The *AO* framework avoids this kind of limits. Now we formalize bridge rules in *AO* framework.

Definition 8 (Mapping Foreign Entity). *Let* r *be a bridge rule of ontology* O_j*, the left part of the* $\rightarrow$ *is a mapping foreign entity of* O_j*.*

Definition 9 (Mapping in *AO* framework). *For a bridge rule* r *in ontology* O_j*, we have in AO*

1. $j{:}i{:}x \subseteq j{:}j{:}y$ *and* $i{:}i{:}x \overset{\equiv}{\Longrightarrow} j{:}i{:}x$, *if* $\mathrm{r}= i{:}i{:}x \xrightarrow{\sqsubseteq} j{:}j{:}y$;
2. $j{:}i{:}x \supseteq j{:}j{:}y$ *and* $i{:}i{:}x \overset{\equiv}{\Longrightarrow} j{:}i{:}x$, *if* $\mathrm{r}= i{:}i{:}x \xrightarrow{\sqsupseteq} j{:}j{:}y$;
3. $j{:}i{:}x = j{:}j{:}y$ *and* $i{:}i{:}x \overset{\equiv}{\Longrightarrow} j{:}i{:}x$, *if* $\mathrm{r}= i{:}i{:}x \xrightarrow{\equiv} j{:}j{:}y$;
4. $j{:}i{:}x \cap j{:}j{:}y = \emptyset$ *and* $i{:}i{:}x \overset{\equiv}{\Longrightarrow} j{:}i{:}x$, *if* $\mathrm{r}= i{:}i{:}x \xrightarrow{\perp} j{:}j{:}y$;
5. $j{:}i{:}x \cap j{:}j{:}y \neq \emptyset$ *and* $i{:}i{:}x \overset{\equiv}{\Longrightarrow} j{:}i{:}x$, *if* $\mathrm{r}= i{:}i{:}x \xrightarrow{*} j{:}j{:}y$.

Table-1 shows that, mappings are formalized into normal language statements together with semantic bindings on the mapping foreign entity in *AO* framework.

5 Conclusions

In this paper, we put forward an *AO framework* of distributed autonomous ontology space, which supports semantical cooperations between distributed knowledge-base.

Firstly we introduce the *AO framework* of autonomous ontology, by which a language entity in an ontology is interpreted locally, nevertheless the semantic cooperation is still keeping. Different from various of works based on DDL (distribute description logic [1]), *AO* framework depends on semantic binding instead of domain-relation to build relationship between different ontologies so that it avoids damaging the autonomy of each local ontology.

Secondly, we discuss on three basic operations among ontologies, say *free-access operation*, *importing operation*, and *mapping operation*. Intuitively, free-access operation allows the ontology space opening to anybody; importing operation allows some ontology to copy other ontology; mapping operation sets up semantical mapping between ontologies. Later we formalize these operations in the AO framework and give out proper semantics to them.

References

1. Borgida, A., Serafini, L.: Distributed description logics: Assimilating information from peer sources. Journal of Data Semantic **1** (2003) 153–184
2. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: C-OWL: Contextualizing ontologies. In: Sencond Internatinal Semantic Web Conference. Volume 2870 of Lecture Notes in Computer Science., Springer Verlag (2003) 164–179
3. Grau, B.C., Parsia, B., Sirin, E.: Combining owl ontologies using e-connections. To appear in Elsevier's Journal Of Web Semantics (JWS) (2005)
4. Grau, B.C., Parsia, B., Sirin, E., Kalyanpur, A.: Modularizing owl ontologies. In: KCAP-2005 Workshop on Ontology Management. (2005)
5. Serafini, L., Tamilin, A.: DRAGO: Distributed reasoning architecture for the semantic web. In Gomez-Perez, A., Euzenat, J., eds.: Proc. of the Second European Semantic Web Conference (ESWC'05). Volume 3532 of Lecture Notes in Computer Science., Springer-Verlag (2005) 361–376
6. Patel-Schneider, P.F., Hayes, P., Horrocks, I.: Owl web ontology language semantics and abstract syntax (2004) W3C Recommendation. `http://www.w3.org/TR/2004/REC-owl-semantics-20040210/`. Latest version available at `http://www.w3.org/TR/owl-semantics/`.
7. Berners-Lee, T., Fielding, R., Masinter, L.: Uniform resource identifiers (uri): Generic syntax. RFC 2396 (1998) http://www.ietf.org/rfc/rfc2396.txt.

SemreX: A Semantic Peer-to-Peer System for Literature Documents Retrieval*

Hai Jin, Hanhua Chen, and Xiaomin Ning

Cluster and Grid Computing Lab
Huazhong University of Science and Technology, Wuhan, 430074, China
hjin@hust.edu.cn

Abstract. The decentralized structure together with the features of self-organization and fault-tolerance makes peer-to-peer network a promising model for information sharing. However, efficient content-based searching remains a serious challenge of large scale peer-to-peer network. In this paper, we present SemreX, a peer-to-peer system for sharing literature documents. Two main features of SemreX networks are 1) semantic supported literature document retrieval function is provided and 2) peers are self-organized into a semantic overlay according to the similarity of documents which belongs to different topics and queries are routed to semantically similar peers to reduce messages. Experiment results show that SemreX improves search efficiency for literature document retrieval in peer-to-peer network.

1 Introduction

Due to characterizes of low maintenance overhead, improved scalability and reliability, synergistic performance, increased autonomy, privacy, and dynamism of peer-to-peer systems, they have shown a great potential on file sharing in recent years and are used by millions of users for file sharing over the Internet [3].

Much effort in peer-to-peer area is made in the research of "title-based" search for file-sharing. Current peer-to-peer search mechanisms can be classified into three types [23]. In the first kind, a centralized index is maintained at a server, and all queries are directed to the server. The centralized index server becomes a performance bottleneck and single point of failure in large scale peer-to-peer systems. Second kind of approach is the *Distributed Hash Table* (DHT) based scheme [18, 24]. Though extremely robust and scalable, these systems suffer from simplistic data models, which consist of collections of key-value pairs, and the exactly-matching based retrieval is not suitable for content search. Although a few works have been aimed at providing a partial-match lookup capability on DHTs [25-27], the inherent characteristics of DHT make these methods sophisticated and much less efficient than the web search engines that are popular for navigating the Internet. Another kind of peer-to-peer networks are commonly called unstructured overlays. Queries randomly walk [28] or are flooded across the network. Generally, flooding based approaches

* This paper is supported by the National 973 Key Basic Research Program under grant No.2003CB317003.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 653 – 667, 2006.
© Springer-Verlag Berlin Heidelberg 2006

may lead to heavy network traffics by generating an exponential number of query messages while random walk methods may reduce the messages at the cost of recall rate. Traditional peer-to-peer overlays fail to support efficient content based information locating in large scale peer-to-peer networks.

Despite the relatively much effort in the research of title-based peer-to-peer search facility, very limited work in the semantic-based content search has been specifically addressed. Notable exceptions are Edutella [9], Bibster [10, 11] and pSearch [20, 21]. Edutella and Bibster propose RDF metadata models that standardize the way data and services are organized and queried in a peer. pSearch distributes document indices through the peer-to-peer network based on document semantics generated by LSI and is organized as a CAN [18]. The search cost for a given query is reduced, since the indices of semantically related documents are likely to be co-located in the network. Although the pSearch approach works well for finding documents close to a query, its performance under highly dynamic peer-to-peer systems is unknown.

In this paper, we introduce SemreX[1], a system for desktop literature documents sharing in peer-to-peer environments. Two main features of SemreX are 1) semantic supported literature document retrieval function is provided and 2) peers are self-organized into a semantic overlay according to the similarity of documents which belongs to different topics and queries are routed to semantically similar peers to reduce messages. SemreX considers the scenario that research participants in computer science share articles in their desktop file systems and the participants will be able to retrieve from SemreX the scientific articles he wants, shown in Fig.1. For example, a user can issue semantic based queries, “Find me a paper with the title including the word ‘ontology’, published in 2005, and about the topic ‘knowledge representation’”. The GUI formulates the query sentence, issues it to the semantic overlay and processes the possible results for the user. Different kinds of data are extracted from the documents on a peer, and the documents are classified into different ACM topics. Experiment results show that SemreX improves search efficiency for literature document retrieval in unstructured peer-to-peer network.

The rest of this paper is organized as follows. In section 2, we describe the system architecture of SemreX. We propose the semantic overlay in section 3. In section 4 we evaluate the system performance. Section 5 reviews some related works. Section 6 concludes the paper and describes our future work.

2 System Architecture of SemreX

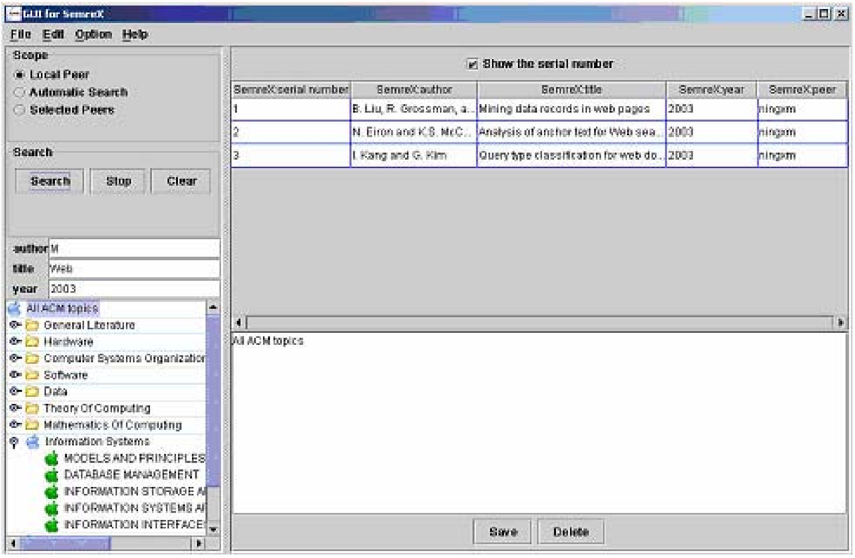
The architecture of SemreX is illustrated in Fig. 2. SemreX consists of five components, including the user interface, the document retrieval component, the semantic management component, the system control component, and the JXTA-based peer communication component. The responsibilities of each component are also specified in the figure.

2.1 Semantic Management Component

Content based information sharing needs the peer information to be managed in more proper orders and to be more efficiently retrieved. The semantic management

[1] http://grid.hust.edu.cn/semrex/

component of SemreX aims at providing homogeneous ontology over the distributed dynamic peer-to-peer network. Metadata of literature documents in local peers is extracted and organized according to the SemreX:Publication ontology (Fig. 3) [15].



● Local Peer
○ Automatic Search
○ Selected Peers

author M
title Web
year 2003

☑ Show the serial number

SemreX:serial number	SemreX:author	SemreX:title	SemreX:year	SemreX:peer
1	B. Liu, R. Grossman, a...	Mining data records in web pages	2003	ningxm
2	N. Eiron and K.S. McC...	Analysis of anchor text for Web sea...	2003	ningxm
3	I. Kang and G. Kim	Query type classification for web do...	2003	ningxm

All ACM topics
- General Literature
- Hardware
- Computer Systems Organization
- Software
- Data
- Theory Of Computing
- Mathematics Of Computing
- Information Systems
  - MODELS AND PRINCIPLES
  - DATABASE MANAGEMENT
  - INFORMATION STORAGE A
  - INFORMATION SYSTEMS A
  - INFORMATION INTERFACE

All ACM topics

Fig. 1. SemreX GUI

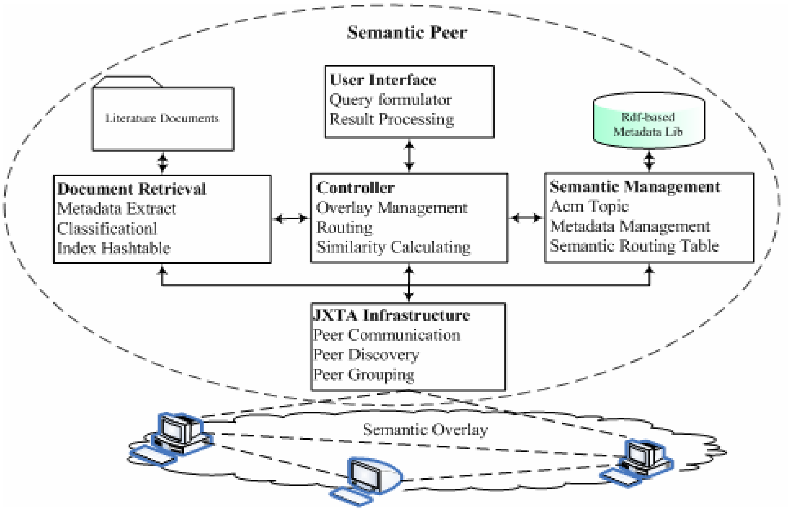


Fig. 2. System architecture of SemreX

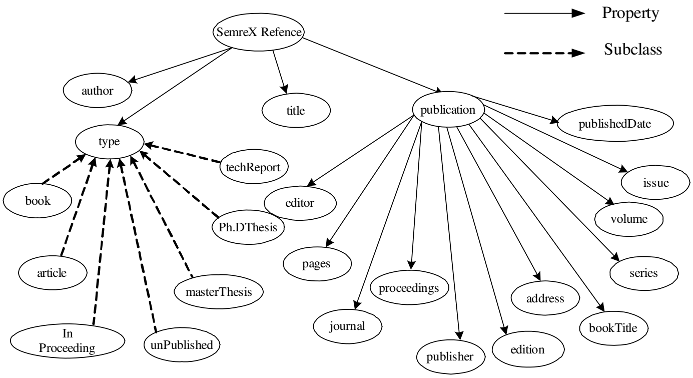


Fig. 3. Publication ontology in SemreX

Local literature documents of computer science are classified according to ACM Topic (seen in Fig. 4) ontology [14], which has become a standard ontology for categorizing computer science literatures and covers 1475 topics of the computer science domain. With sub- and super topic hierarchy the concepts are associated in the IS-A structure.

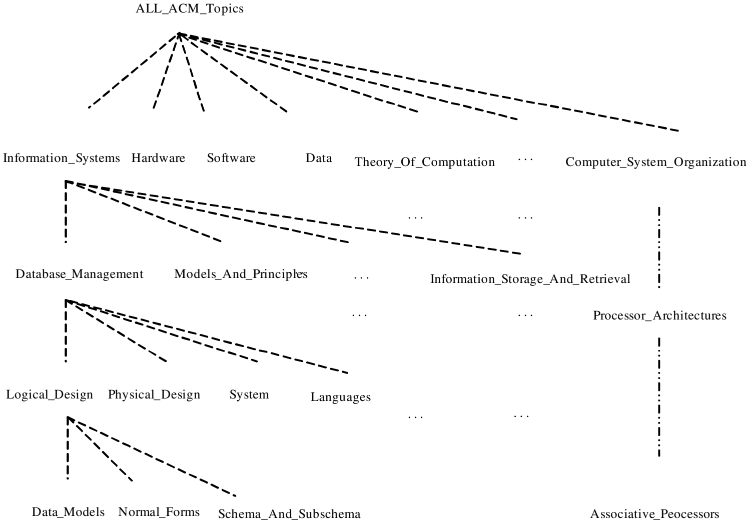


Fig. 4. ACM topic ontology IS-A concept structure

These two ontologies bridge the gap between the views of SemreX users and the distributed knowledge source. Furthermore, the ACM Topic ontology contributes to the semantic based overlay of SemreX and improves query routing across the peer-to-peer networks (detailed in section 3).

2.2 Document Retrieval Component

Document retrieval component views local peer as a set of literature documents in heterogeneous formats, for example pdf, ps, and etc. Peers in the distributed network share the aforementioned homogeneous ontology. Documents on a peer are classified into different ACM topics, and thus each peer's semantic is represented by these topics with statistical proportion.

$$P =\{ d_j , j=1,2,...,n\}=\{<T_i, \lambda_i>, i=1,2,..,m\} \qquad (1)$$

Here, n denotes the number of document distributed in a peer and m denotes the number of topics in the peer. We use the following matrix $C(P)$ to describe the relationship of the documents and the topics in a peer:

$$C(P) = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix} \qquad (2)$$

where c_{ij} is defined as following:

$$c_{ij} = \begin{cases} 1 & if\ d_j \in T_i \\ 0 & if\ d_j \notin T_i \end{cases} \qquad (3)$$

In equation (1), λ_j is the proportion of the statistical occurrence of documents belonging to T_i. It is calculated by the following method:

$$\lambda_i = \frac{\sum_{j=1}^{n} c_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}} = \frac{\sum_{j=1}^{n} c_{ij}}{n} \qquad (4)$$

For document classification, we apply *Latent Semantic Indexing* (LSI) [6, 8] in information retrieval to reveal semantic subspaces of feature spaces from documents stored on peers. After producing semantic vectors through LSI, we train a *support vector machine* (SVM) [7] to classify the peer documents into different categories based on the extracted vectors. Supervised classification using SVM involves a training phase and a prediction phase. During the training phase, a large set of documents with known category labels are used to train the classifier. During classification, we flat the ACM Topic tree into a topic collection for classification. Detail information about literature document classification can be found in our previous paper [16].

In order to manage local documents, each peer maintains a local index containing a hash table for mapping the global identify of any document on the peer-to-peer system to the physical file location in the local file system.

2.3 Peer Controller

In SemreX network, overlay management and query routing are significant. In the scientific literature retrieval scenario, all participant researchers are self-organized into semantic peer-to-peer overlay according to their *research interests*. In the peer controller component design a local peer is described as $P =\{<T_i, \lambda_i>, i=1,2,..,m\}$, where $\{T_i\}$ denotes the research topics that the peer owner is interested in, and λ_i shows how much he is interested in T_i (Equation 4). Different from other self-organized overlay, the basic idea of SemreX semantic overlay is to cluster the peers which have similar topics and the main idea of query routing in SemreX semantic overlay is to forward the queries to the peers which have similar topics with the semantic of the queries and are most possible to return results. The semantic clustering strategy is based on the semantic similarity between peers. We will describe the semantic overlay in detail in section 3.

2.4 Peer Communication Component

The peer communication component serves as a transport layer for other components of SemreX and hides all low-level communication details from the rest components. In the specific implementation of SemreX system we use JXTA as the communication platform. Our previous work has described the JXTA based communication in SemreX in detail [17].

3 SemreX Overlay

3.1 Semantic Similarity Based Overlay

Definition 1. Semantic similarity of peers in SemreX is defined as the semantic similarity between the corresponding pair of sets of weighted topics, which are concept nodes on the ACM Topic IS-A concept structure.

$$Sim(P^1, P^2)=Sim(\{<T_i, \lambda_i>, i=1,2,..,m\}, \{<T_j, \lambda_j>, j=1,2,..,n\}) \qquad (5)$$

The study of semantic similarity between lexically expressed concepts has been a part of natural language processing and information retrieval for many years [1]. A number of semantic similarity methods have been developed in the previous decade, and different similarity measure methods have proven to be useful in some specific applications of computational intelligence.

Generally, these methods can be categorized into two groups: edge counting- based (or dictionary/thesaurus-based) methods and information theory-based (or corpus-based) methods.

The first group relates the concept similarity to the minimal path length [4] and the depth of the subsumer of the two concepts in the concept structure. Among this kind of methods, L. Yuhua's measure [12] gives the best results.

$$Sim(T_1,T_2) = f_1(l)\cdot f_2(h) = \begin{cases} e^{\alpha l}\cdot \dfrac{e^{\beta h}-e^{-\beta h}}{e^{\beta h}+e^{-\beta h}} & if(T_1 \neq T_2) \\ 1 & if(T_1 = T_2) \end{cases} \qquad (6)$$

Here, l counts the shortest path length between T_1 and T_2 and h counts the depth from the subsumer of T_1 and T_2 to the top of the concept hierarchy. $\alpha \geq 0$ and $\beta > 0$ are parameters scaling the contribution of shortest path length and depth, respectively. The strongest correlation between formula (6) and human judgments is at 0.2 and 0.6. Using this kind of measure the similarity is statically decided by the concept structure.

The basic idea of the corpus-based methods [13] is that the more information two concepts share in common, the more similar they are, and the information shared by two concepts is indicated by the information content of the concepts that subsume them in the taxonomy [5].

$$Sim(T_1, T_2) = \underset{T \in S(T_1, T_2)}{Max} [-\log p(T)] = -\log p[lso(T_1, T_2)] \tag{7}$$

Here, T_1 and T_2 are two topics in the concept tree, and $S(T_1,T_2)$ is the set of concepts that subsume T_1 and T_2, that is to say T is a 'super-class' of T_1 and T_2 in common. $p(T)$ denotes the probability that concept T occurs in the corpus. $p(T)$ is quantified as $p(T) = \frac{freq(T)}{N}$, where $freq(T)$ quantifies the times that topic T occurs and N denotes the sum of occurrence times of all topics. As occurrence times of any topic will add to that of its super-topic, the formula (7) implies that -log $p(T)$ is monotonically non-decreasing as T moves up the concept tree. So, the information content of the T reaches the maximal value when $T = lso(T_1,T_2)$, which is the most specific super-class among $S(T_1,T_2)$ in the concept tree.

In this paper we use method in [12] and the similarity between two peers is quantified as follow:

$$Sim(P^1, P^2) = \sum_{j=1}^{j \leq |P^1|} \sum_{i=1}^{i \leq |P^2|} [Sim(T_i, T_j) \times (\lambda_i \times \lambda_j)] \tag{8}$$

Here, $|P^1|$ and $|P^2|$ are the topic numbers in the two peers. λ is calculated by equation (4). The similarity between the sets of weighted concepts of each other is calculated by summing up products of the similarity value between any pair of topics separately selected from P^1, P^2 and the weights of both topics.

Based on the above method to measure the similarity of peers, we come to the algorithm for generating the semantic overlay. First, we use the following expression to describe the neighbors of peer P^k:

$$Neighbor_{semantic}(P^k) = \{P^k_1, P^k_2, \ldots, P^k_m\} \quad m>0 \tag{9}$$

The basic idea of SemreX semantic overlay is to cluster the peers which have similar topics. When a peer enters the network it advertises its semantic description in the network. Other peers will decide to accept the new comer as a neighbor or not based on the degree of semantic similarity between them. Algorithm 1 specifies the detail of the strategy for generating semantic overlay.

```
Algorithm 1. Semantic Overlay Generation
//Assume current peer is $P^j$
Input:   $SemreX=\{P^1,P^2,\ldots,P^n\}$,
         $P^k=\{<T^k_i, \lambda^k_i>, i=1,2,..,m^k\}$ is any peer in the network
Output:  $Neighbor_{semantic}(P^j)$
Procedure:
set $Neighbor_{semantic}(P^j)= \Phi$ ;
set $P^j.TTL=TTL_0$;
advertise ( $\{<T^j_i, \lambda^j_i>, i=1,2,..,m^j\}$ and $P^j.TTL$);
while(true) do
    received semantic advertise from peer $P^k$ ;
    $P^k .TTL= P^k .TTL -1$.
    if ($Sim(P^j, P^k) > Threshold_{peer\_sim}$) do
        set $Neighbor_{semantic}(P^j)= Neighbor_{semantic}(P^j) \cup \{ P^k \}$;
    end if
    if ($P^k .TTL>0$) do
        forward advertise ( $\{<T^k_i, \lambda^k_i>, i=1,2,..,m^k\}$ and $P^k.TTL$);
    end if
end do
```

3.2 Query Routing in SemreX Overlay

The search strategy of SemreX is based on the measure of the similarity between a query and a peer.

Definition 2. Similarity between a query and a peer in SemreX is quantified as the maximal value of all the products of the similarity value between the topic about which the query is to search and any topic in peer and the weight of the topic from the peer.

$$Sim(T_Q, P) = \max_{T_i \in P} \left\{ Sim(T_Q, T_i) \times \lambda_i \right\} \tag{10}$$

Here, T_Q is the topic about which the query is to search. P is the peer to be compared with. λ_i is the weight of topic T_i in the peer P.

The basic idea of query routing strategy in SemreX is to route the query messages to the peers which are much more similar with the queries and are mostly like to return the results. When any peer receives a query message, it calculates the similarity between the query and the semantic representation of the peers in the routing table and selects similar peers to forward the query.

$$Sim(T_Q, P) > Threshold_{semantic\_similarity} \tag{11}$$

For this method there is a potential “danger of swamp”, that is to say no similar candidates are available and the search process is forced to stop before getting any results. To solve this problem, we here introduce a random mechanism for the query to “jump out of the swamp peers”. In this case, the peer P^l is selected to forward the message at the probability decided by equation (12):

$$p_{forword}(P^l)=\begin{cases}\dfrac{Sim(T_Q,P^l)-Sim_{\min}}{Sim_{\max}-Sim_{\min}} & if\,(Sim_{\max}-Sim_{\min}\neq 0)\\ 1 & if\,(Sim_{\max}-Sim_{\min}=0)\end{cases} \tag{12}$$

where Sim_{max} is the similarity value of the most similar neighbor, and Sim_{min} the least similar neighbor peer.

Detail of the semantic query routing strategy is specified in Algorithm 2. The algorithm focuses on the semantic queries routing, and the simple keyword matching based queries can be processed by broadcast or sending to a random set of neighbors.

Algorithm 2. Semantic Query Routing

```
Input: $Q=Query(Exprssion(T_Q); P^k.id; TTL)$, $P^k$ is the source peer.
        $P^j=\{<T^j_i, \lambda^j_i>, i=1,2,..,m^j\}$
        $Neighbor_{semantic}(P^j)= \{P^j_1,...,P^j_s\}\ \ s>0$
Output:
        Result $R$ to $Q$.
        Neighbor peer collection $P_{sim}(Q)=\{P_1,P_2,...,P_r\}$ to route.
Procedure:
    set $P_{sim}(Q)= \Phi$;
    while(true) do
        listening query messages from the network
        if ($Query(Exprssion(T_Q); P^k.id; Q.TTL)$ from $P^k$ received)
            $Q.TTL= Q.TTL-1;$
            if ($Sim(T_Q, P_j)$> $Threshold$)
                performing local search and return result $R$;
            end if
            if ($Q.TTL>0$)
            Boolean Flag_swamp=True;
            for each peer $p^x$□$Neighbor_{semantic}(P^j)\ 1\leq i\leq m$ do
                if ($Sim(T_Q, p^x)$> $Threshold$)
                    $P_{sim}(Q)= P_{sim}(Q)\backslash\psi p^x \}$;
                    if(Flag_Swamp=True)
                        Flag_swamp=False;
                    end if
                end if
              end for
              if(Flag_swamp=True) /*jump out of swamp*/
                  for each peer $p^x$□$Neighbor_{semantic}(P^j)\ 1\leq i\leq m$ do
                      generate a random number r, where 0<r<1
                      if ($r < p_{foward}(p^x)$)
                          $P_{sim}(Q)= P_{sim}(Q)\backslash\psi p^x \}$;
                      end if
                  end for
              end if
              forward $Q$ to every peer in collection $P_{sim}(Q)$;
        end if
    end do
```

4 Experiment Results

4.1 System Evaluation

The area of peer-to-peer information retrieval is rather new and there are no established standard evaluation functions. The implementation of SemreX presented in this paper will be evaluated by means of a study among the potential end users of the system. Although we cannot report the results of real environment with large number of users at the time of writing this paper, we simulate SemreX semantic overlay and present our simulation result in this section. In the simulation experiments, we focus on the efficiency of the semantic overlay. The experiment consists of two aspects: (1) evaluating the similarity of topics based on document classification using SVM, and (2) evaluating the search efficiency of semantic overlay compared with that in Gnutella.

4.2 Evaluation of Similarity Based on Document Classification

We use SVM to classify documents of computer science. In this experiment, we use a large set of abstracts from Computer Science Database. We choose abstracts of three top levels of ACM topics: *C.Computer Systems Organization*, *G.Mathematics of Computing*, and *H.Information Systems*. Each category contains approximately 1000 abstracts for training and 100 abstracts for test. The final training data contains 3309 abstracts and the corpus encloses 12582 words. Using optimal parameters, the SVM model predicts 300 documents. The average accuracy of the classification is 89.6%**.** Performance metric accuracy describes the proportion between the correctly classified documents and all documents to classify. Based on the document classification we test the similarity between topics. Table 1 shows examples of the value of topic similarity using formula (**6**).

Table 1. Similarity between topics

Topic 1	Topic 2	Similarity
Distributed systems	High-Speed networks	0.519
Client/server	Network operating systems	0.659
Distributed database	Data models	0.132
Routers	Super computers	0.251
Digital library	Information search and retrieval	0.634
Linguistic processing	Associative processors	0.132
Formal languages	Pattern matching	0.306
Data models	Information search and retrieval	0.306
Algebraic language theory	Formal languages	0.775
Content analysis and indexing	Retrieval models	0.519

4.3 Evaluation of Semantic Overlay

This experiment focuses on evaluating the recall rate, search cost and the efficiency of SemreX semantic overlay. Recall is a standard measure in information retrieval field. It describes the proportion of all relevant documents included in the retrieved set.

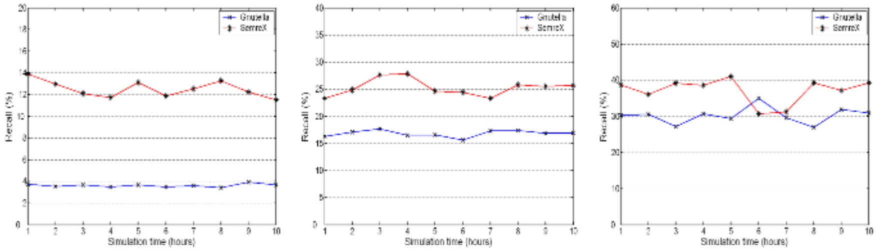
$$Recall = \frac{|\,Document_{relevant} \bigcap Document_{retrieved}\,|}{|\,Document_{relevant}\,|} \tag{13}$$

Generally speaking, the most notable overhead in peer-to-peer systems tends to be the processing cost for queries. Search cost is average number of messages caused by per search request. Efficiency is the ratio of recall and search cost. To have a fair comparison among different metrics, this metric considers both recall and search cost together and is to measure the overall performance.

Table 2. Settings for evaluating recalls and traffic

Parameters	Parameter Descriptions	Values
N	Number of nodes in the network	1k – 4k
α	Exponent α of power law	3.0
d	Average degree	2.8-3.4
T	Number of ACM topics in the network	30
D	Max number of documents on each node	500
Q	Max number of queries by each peer	200
C	Number of keywords in each document	1-50
TTL	Time to Live for searching	2 - 4

In the experiment, the simulator generates semantic overlay from the original Gnutella-like network. Four original graphs with different scales are used as original Gnutella-like [2] networks in simulation. Each of them accords with a *power-law* with the exponent *α* = *3.0* and the average degree 2.8~3.4.



(a) *TTL*=2 (b) *TTL*=3 (c) *TTL*=4

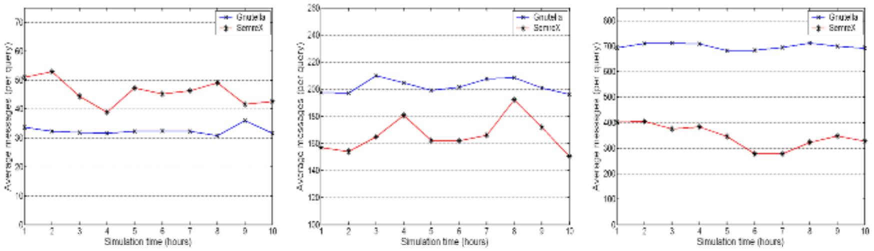
Fig. 5. Comparing recall when varying TTL values

The population of documents follows a Zipf distribution with parameters α = 1.2 and n = 500, as studies have shown that the file popularity distribution in Kazaa follows Zipf's law [22]. The distribution of search requests issued by each peer accords with a Zipf distribution with α = 1.0 and n = 200. Table 2 summarizes the simulation settings.

We compare the efficiency of the semantic overlay and the Gnutella style. Fig. 5 shows that the recall rates of both our semantic overlay model and Gnutella increase as the *TTL* increases when the scale of simulation network is fixed at 1000. The recall rate of our model outperforms Gnutella apparently.

We also note that the recall rates of SemreX overlay have less advantage when *TTL* equals 4. The probable reason is that the number of nodes in our experiment is relatively small and the average shortest distance is rather small. When we set *TTL* = 4, the Gnutella style can crawl most nodes of the network but causes very heavy traffic at the same time.

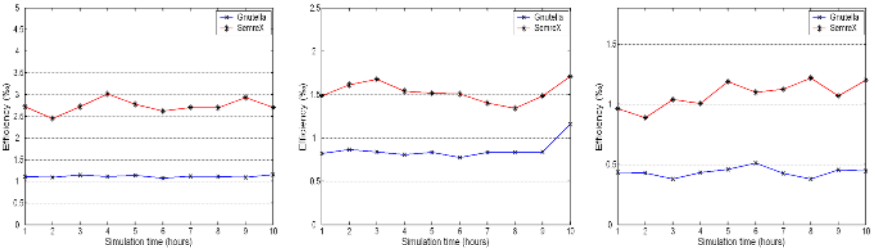
With fixed scale of 1000 nodes, we change *TTL* from 2 to 4. Simulation result shows that our model can effectively reduce message traffic as *TTL* increases, shown in Fig. 6.



(a) *TTL*=2 (b) *TTL*=3 (c) *TTL*=4

Fig. 6. Comparing average messages per query request when varying TTL values

Figure 7 shows that SemreX model can increase the searching efficiency dramatically. We extend the scale of simulation networks from 1000 to 4000. Figure 8 shows the recall rate of our model greatly outperforms Gnutella in every scale and the advantage of SemreX increases apparently as the scale of the peer-to-peer network increases.



(a) *TTL*=2 (b) *TTL*=3 (c) *TTL*=4

Fig. 7. Comparing search efficiency when varying TTL values

The simulation results show that the query routing algorithm in the semantic overlay increases the recall rate and reduces message traffic dramatically.

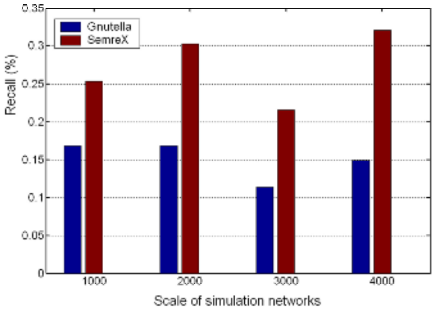


Fig. 8. Comparing recall when varying the scale of nodes from 1000 to 4000

5 Related Works

The area of peer-to-peer information retrieval is rather new, and very limited work has been done. The proposed systems can be categorized into IR over *structured* peer-to-peer systems [19], such as pSearch and IR over *unstructured* peer-to-peer systems, such as bibster.

pSearch aims at forming a overlay networks for distributing document indices through the peer-to-peer network. The overlay is based on document semantics generated by LSI and organized as a CAN. Although the pSearch approach works well for finding documents close to a query, its performance under the dynamic conditions, which are common for peer-to-peer systems, is unknown.

Bibster is another model based on semantic overlay. It is the most related work to ours. In Bibster, the knowledge about the expertise of other peers forms a semantic topology and the expertise of peers is extracted from documents just using lexical analysis and leads to many false categorization results (only approximately 10% accuracy according to our evaluation on blister), so the semantic topology based on the expertise in Bibster is not dependable. By the time when this paper is written, we got no published evaluated results of Bibster in large scale peer-to-peer networks.

6 Conclusion and Future Works

In this paper we present SemreX, a system for desktop literature documents sharing in peer-to-peer network environments. We present a semantic overlay algorithm with that semantically similar peers are clustered together. A query can be efficiently routed to semantically similar peers and thus the recall rate is increased while the message overhead of the peer-to-peer overlay is reduced.

In the next step, we will consider evaluating the semantic overlay and querying routing algorithms in networks with much larger scales, and try to find statistical properties of the generated semantic overlay, such as average path lengths, degree distributions, and cluster coefficient that characterize the structure of the network. We will also consider evaluating SemreX system by means of a study among the potential end users of the system and report the result in subsequent report.

References

[1] Budanitsky and G. Hirst, "Semantic Distance in WordNet: an Experimental, Application-oriented Evaluation of Five Measures", *Proceedings of workshop WordNet and Other Lexical Resources*, June, 2001.

[2] Gnutella, http://gnutella.wego.com/, 2000.

[3] Napster, http://www.napster.com, 2001.

[4] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and Application of a Metric on Semantic Nets", *IEEE Transactions on System, Man, and Cybernetics*, Vol.19, No.1, Jan. 1989, pp.17-30.

[5] P. Resnik, "Semantic Similarity in a Taxonomy: an Information-Based Measure and its Application to Problems of Ambiguity in Natural Language", *Journal of Artificial Intelligence Research*. Vol.11, 1999, pp.95-130.

[6] M. S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis", *Journal of American Society for Information Science*, Vol.41, No.6, 1990, pp.391-407.

[7] E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers", *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, ACM Press, July, 1992, Pittsburgh, PA, pp.144-152.

[8] M. W. Berry, Z. Drmac, and E. R. Jessup, "Matrices, Vector Spaces, and Information Retrieval", *SIAM Review*, Vol.41, No.2, 1999, pp.335-362.

[9] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch, "Edutella: a Peer-to-Peer Networking Infrastructure Based on RDF", *Proceedings of WWW'02*, Hawaii, USA, May 2002, pp.604-615.

[10] P. Haase, J. Broekstra, M. Ehrig, M. Menken, P. Mika, M. Plechawski, P. Pyszlak, B. Schnizler, R. Siebes, S. Staab, and C. Tempich, "Bibster: A Semantic-Based Bibliographic Peer-to-Peer System", *Proceedings of ISWC 2004*, *LNCS*, Vol.3298, Springer-Verlag, Nov. 2004, pp.122-136.

[11] P. Haase and R. Siebes, "Peer Selection in Peer-to-Peer Networks with Semantic Topologies", *LNCS*, Vol. 3226, Springer-Verlag, June 2004, pp.108-125.

[12] L. Yuhua, Z. A. Bandar, and D. McLean, "An Approach for Measuring Semantic Similarity Between Words Using Multiple Information Sources", *IEEE Transactions on knowledge and data engineering*, Vol.15, No.4, July/August 2003, pp.871-882.

[13] J. J. Jiang and D. W. Conrath, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy", *Proceedings* of *International Conference Research on Computational Linguistics (POCLING X)*, 1997, Taiwan.

[14] The ACM Topic Hierarchy. http://www.acm.org/class/1998.

[15] Z. Guo, H. Jin, and H. Chen, "Semantic Information Extraction of Reference Metadata in SemreX", *Journal of Computer Research and Development. 2006.*

[16] H. Jin, X. Ning, and H. Chen, "Efficient Query Routing in Semantic Overlays Based on Latent Semantic Indexing", *Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC'06)*, Dijon, France, April 23-27, 2006.

[17] Y. Yu and H. Jin, "Building a Semantic P2P Scientific References Sharing System with JXTA", *LNCS*, Vol.3841, Springer-Verlag, 2005, pp.937-942.

[18] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. "A Scalable Content-Addressable Network", *Proceedings of ACM SIGCOMM'01*, San Diego, California, USA, Aug. 2001.

[19] K. Aberer, F. Klemm, M. Rajman, and J. Wu, "An Architecture for Peer-to-Peer Information Retrieval", *Proceedings of 27th Annual International ACM SIGIR Conference Workshop on P2PIR*, July 29, 2004, pp.32-42.

[20] Z. Zheng, M. Mallik, Z. Xu, and W. Tang, “On Scaling Latent Semantic Indexing for Large Peer-to-Peer Systems”, *Proceedings of 27th Annual International ACM SIGIR Conference*, Sheffield, UK, July, 2004, pp.112-121.

[21] Sujata, Z. Xu, and S. Dwarkadas, “Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks”, *Proceedings of ACM SIGCOMM’03*, Karlsruhe, Germany, Aug. 2003, pp.175-186.

[22] Iamnitchi, M. Ripeanu, and I. Foster, “Small-world file-sharing communities”, *Proceedings of IEEE INFOCOM'04*, Hong Kong, 2004.

[23] H. T. Shen, Y. Shu, and B. Yu, “Efficient semantic-based content search in p2p network”, *IEEE Transactions on Knowledge and Data Engineering*, Vol.16, No.7, July, 2004, pp.813-826.

[24] Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: a scalable peer-to-peer lookup service for internet application”, *Proceedings of ACM SIGCOMM'01*, San Diego, California, USA, 2001.

[25] P. Reynolds and A. Vahdat, “Efficient peer-to-peer keyword searching”, *Proceedings of Middleware*, 2003.

[26] O. D. Gnawali, “A Keyword-Set search system for peer-to-peer networks”, Master's thesis, Massachusetts Institute of Technology, June, 2002.

[27] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer, “P2p content search: give the web back to the people”, *Proceedings of the 5th International Workshop on Peer-to-Peer Systems (IPTPS'06)*, Santa Barbara, CA, USA, 2006.

[28] Gkantsidis, M. Mihail, and A. Saberi, “Random walks in peer-to-peer networks”, *Proceedings of IEEE INFOCOM'04*, Hong Kong, China, 2004.

Personal Information Modeling in Semantic Web

Sabah S. Al-Fedaghi and Majed Y. Ahmad

Computer Engineering Department
Kuwait University
P. O. Box 5969 Safat 13050
Kuwait
sabah@eng.kuniv.edu.kw

Abstract. Web Ontology Language (OWL) is a semantic markup language for describing information on the Web so that machines can process and interpret Web content. OWL expresses information ontologies and has the capability to map all the information on the World Wide Web into a semantic, machine-understandable atlas of information. This article addresses personal information modeling in OWL through enhancing the OWL specification to include such information. It introduces a scheme for identifying personal information and presents an OWL-based design to model personal information without introducing modifications. Some restrictions on certain OWL constructs are necessary to integrate personal information in the language.

1 Introduction

The notion of privacy is becoming an important feature in all aspects of modern society. According to Gleick [8], “[P]rivacy will be to the information economy of the next century what consumer protection and environmental concerns have been to the industrial society of the 20th century.” The fast pace of advances in information and communication technology has contributed to greater concern about privacy.

First, privacy is related to an individual’s concern about privacy erosion. In many cases, the disclosure of personal behavior and information causes embarrassment, even when there is no blame attached to the action [2]. Protection of privacy is also necessary against inappropriate and unlawful uses of personal information (e.g., identity theft). Nevertheless, the appetite for personal information is increasing in all aspects of life. Today's information technologies include surveillance tools, such as video, networks, global positioning systems, black boxes in cars, genetic testing, biometric identifiers, and radio frequency identification devices (RFIDs). The 2005 report of the Privacy Commissioner of Canada states the following:

> New technologies designed for, or capable of, surveillance of individuals are widespread and are used not only by law enforcement and national security agencies. Businesses, individuals — even your new car — are gathering personal data using surveillance cameras, spyware, infrared heat sensors and data mining, often without your knowledge or consent. Personal information has become a lucrative commodity … integrated information systems …

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 668 – 681, 2006.
© Springer-Verlag Berlin Heidelberg 2006

> collect and analyze significant amounts of personal information about our travel patterns, financial transactions, and even in some cases the people with whom we associate. The systems analyze and mine the personal data in an attempt to find patterns that might suggest an individual is a security threat, a money launderer, or is financing a terrorist group [15].

Second, there is the threat of the government:

> Government is collecting, analyzing, and sharing more personal information, helped along by improved technology, new legislation, government reorganization, and greater co-operation with foreign states. Flows of personal information are likely to have increased significantly among government departments and agencies…
>
> As law enforcement and national security agencies collect more information, from more sources about more individuals, the probability increases that authorities will make decisions based on information of questionable accuracy or take information out of context. Misuse, misinterpretation, or improper disclosures of personal information can have serious adverse consequences for individuals, families, and even communities…
>
> [The demands of e-government]:
>
> [G]overnment on-line may demand interoperable systems that pool personal information and make it available to more users for more purposes. The greater the amount of information, access, and number of users, the greater the vulnerability of the individuals to excessive government or bureaucratic surveillance [13].

Third, privacy is related to transborder data flows. Globalization "means not just international trade in goods; it also means an extensive traffic in personal information for off-shore processing and storage by both governments and the private sector" [15]. This implies that privacy is not only a "local problem," but also concerns international parties, which hold and process personal information.

Several distinct types of privacy have been distinguished, such as "physical privacy," privacy of personal behavior, privacy of personal communications, and privacy of personal data. For example, physical privacy is described as the quality or state of being apart from bodily interactions (i.e., freedom from sensory interference or intrusion through bodily interactions) [3], [4]. In this paper we concentrate our discussion on informational privacy. Informational privacy concerns personal information [1].

Personal information has become the "basic fuel" for modern businesses and governments to carry out their many services effectively [13]. The central component of "nearly all definitions of information privacy is the term 'personal information'" [11]. "Personal information" is said to denote information about identifiable individuals in accessible form [16]. Defining personal information as "information identifiable to the individual" does not mean that the information is "especially sensitive, private, or embarrassing. Rather, it describes a relationship between the information and a person, namely that the information—whether sensitive or trivial—is somehow identifiable to an individual" [11].

This paper aims at developing an OWL Lite model that can be used for the purpose of analyzing and classifying personal information in order to facilitate automatic exchange of such information. We introduce "personal information ontology" in OWL that includes the categories of personal information existing in the privacy domain in order to produce a catalog that details the types of pieces of information and their relationships relevant for privacy. Personal information representation in OWL is accomplished through mapping personal information statements to the OWL construct while minimizing restrictions on these constructs.

2 Privacy and the Semantic Web

Privacy concerns are increasingly important in the World Wide Web environment, where controlling the creation, gathering, processing, and disclosing of personal data has become a widely discussed issue. Standard privacy policies are not a practical way to keep users informed about how their personal information will be shared. P3P data schemata still lack the expressive power and the semantics required for the definition of different types of personal information that will be described later in this paper.

Semantic Web languages like OWL and RDFS are suitable to represent personal information inasmuch as they integrate structural definitions with a data schema expressing the meaning of the information to be exchanged. RDF and OWL aim at giving meaning to Web information such that machines can understand and process the content of the Web [7], [10]. OWL specifically has the expressive power that can take advantage of ontology-based descriptions of the resources [10].

In OWL, ontologies consist of descriptions of classes, properties, and instances of classes [9]. Using metadata to describe information facilitates the automatic exchange and processing of information[7]. It assists knowledge sharing and exchange through automated processing of Web resources. According to the World Wide Web Consortium, "Using a metadata schema to describe the formal structure of privacy practice descriptions will permit privacy practice data to be used along with other metadata in a query during resource discovery, and will permit a generic software agent to act on privacy metadata using the same techniques as used for other descriptive metadata" [12]. Of special importance in the context of Semantic Web is to automate interactions involving personal information exchange. A fundamental abstraction in achieving this is identifying basic "units" of personal information.

The W3C has proposed utilizing P3P for informational privacy–related context. P3P is a protocol that specifies a Web site's policies with a user's data, such as retention policy, exchange policy, data uses, etc. Its design specifies the syntax and semantics necessary to describe data uses, data recipients, data retention policy, etc. It also includes a standard set of data elements as well as a mechanism for associating policies with Web resources. Sites may also declare additional data elements by publishing their own schemas. However, P3P lacks the significant expressive power available in OWL. It is also not expressed in standardized ontology syntax. If data are to be processed and exchanged freely, then a representation of personal information at the metadata level provides inherent protection, reduces hindrances to information processing, and limits the need for policy level safeguards.

Our work aims at developing a personal information ontology for the purpose of analyzing and classifying personal information in order to facilitate automatic exchange of personal information. “Personal information ontology” refers to the categories of personal information that exist in the privacy domain; thus, the ontology produces a catalog that details the types of pieces of information and their relationships that are relevant for privacy [15].

Currently, there is no explicit distinction between personal information and “owned” information or information of interest to the person. In our ontology of personal information, the system would “recognize” personal information and distinguish it from non-personal information. Agents will be able to recognize that the requested data is personal data (of the agent’s owner or otherwise) and respond accordingly. We claim that our ontological treatment of personal information in the context of OWL is a useful contribution to building privacy into the Semantic Web. According to Kim et al., ontology for building privacy into the Semantic Web is needed now [12].

It also proposes to construct an ontological foundation for modeling identifiable-person types of resources separate from other types of entities. Accordingly, our model consists of “person-resources” represented as nodes that refer to identifiable persons and statements about these persons. One clear advantage of such a model is that there are well-defined nodes of distinct entities: identifiable persons. In general, in our model, resources are divided into two categories: those that represent identifiable persons and those that identify anything else. The basic characteristic of personal information is that it uniquely identifies a REAL person. This person is not an interpretation that depends on namespaces. He/she is a single person who has been or was documented to exist in this world and may have several identities and descriptions. Identifiable human beings are the only “resources” that have this ontological unique identification and are declared as individuals in a special class in OWL called PROPRIETOR class.

3 Personal Information

Personal information is said to denote information about identifiable individuals in accessible forms [16]. Defining personal information as “information identifiable to the individual” does not mean that the information is “especially sensitive, private, or embarrassing. Rather, it describes a relationship between the information and a person, namely that the information—whether sensitive or trivial—is somehow identifiable to an individual” [11]. We adopt the definition of private/personal information (PI) proposed [3], which assumes a universal set of personal information agents of two fundamental types: *Person* and *Nonperson*. *Person* represents the set of natural persons; *Nonperson* represents the set of non-persons. Private/personal information (PI) is any linguistic expression that has referent(s) of type Person.

Definition. *Personal* information is any linguistic expression that has referent(s) of type *Person*. Assume that p(X) is a sentence such that X is the set of its referents. There are two types of personal information*:*

(1) p(X) is atomic personal information if $X \cap V$ is the singleton set $\{x\}$, i.e., atomic personal information is an expression that has a single human referent.
(2) p(X) is compound personal information if $| X \cap V | > 1$, i.e., compound personal information is an expression that has more than one human referent.

In [3], the relationship between persons and their own atomic personal information is called *proprietorship*. If p is a piece of atomic personal information of $v \in V$, then v is its *proprietor*. For example, John is the proprietor of *John is tall*. A *possessor* refers to any agent in $\{$*Person* $\cup$ *Nonperson*$\}$ that knows, stores, or owns the information. People may possess the personal information about a person but they are not the proprietor. And the possessor of the personal information can be either person or non-person (e.g. government agency or a company). Notice that atomic personal information lends itself easily to the tabular form (tables). For example, in the relation STUDENT (NAME, SSN, Grade), a tuple such as (John Smith, 123456678, "excellent") embeds three pieces of atomic personal information: *John Smith is a student*; *his SSN is 123456678*, and *his grade is "excellent."*

Any compound private assertion is privacy-reducible to a set of atomic personal assertions [3]. For example, *John and Mary are in love* can be privacy-reducible to *John and someone are in love* and *Someone and Mary are in love*. Reducing a compound assertion to a set of atomic assertions refers to isolating the privacy aspects of the compound assertion. This means that, if we remove the atomic element from the compound assertion, then the remaining part will not be a "purely" privacy-related assertion with respect to the individual involved. However, it is obvious that privacy-reducibility of a compound personal assertion causes a loss of "semantic equivalence" since the identities of the referents in the original assertion are separated. Semantic equivalency here means preserving the totality of information: the atomic assertions and their link. Suppose that a hospital database includes the information *V1 is V2's kidney donor*. The semantic description would include the two atomic assertions *V1 is a kidney donor* and *V2 had kidney transplantation*. These two assertions can be stored in the two different private databases of V1 and V2. A control mechanism facilitates any access to these facts separately. We can connect the two pieces of personal information by creating a non-personal information link.

4 Personal Information in OWL

In this section we supply the definitions of atomic and compound personal information in OWL Lite.

4.1 Atomic Statements

Atomic personal statements are represented in OWL in a straightforward manner. The proprietor in the atomic statement is an individual related by a property to a value. For example, the statement *John likes apples* can be represented as shown in figure 1. The task of identifying personal information is simply a matter of recognizing the classes whose instances are proprietors. We introduce a new class called *Proprietor*. Any individual that refers to an identifiable person must be an instance of this new OWL class. This will not disturb the meaning of the ontology because OWL allows an individual to be in two classes or more at the same time[5]. This is accomplished by declaring the individual class equivalent to the *Proprietor* class using the OWL construct owl:equivalentClass. Such a technique would prevent statements that include individuals of a *person* class who are not uniquely identifiable or are fictitious characters from being counted as personal information.

Example. the statement *John likes apples* is represented as shown in figure 1.

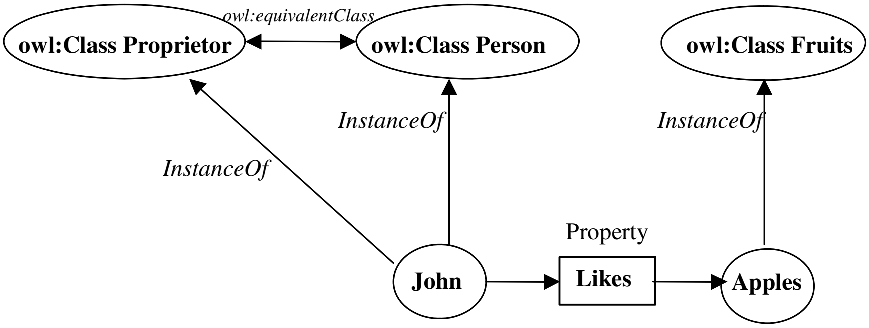


Fig. 1. Atomic Information in OWL

The simple ontology shown in the figure could be inferred from the following OWL declarations:

```
...
<owl:Class rdf:ID="Person"/>
<owl: Class rdf:ID="Fruits"/>
<Fruits rdf:ID="Apples" />
<Person rdf:ID="John" />
<owl:ObjectProperty rdf:ID="likes"/>
<Person rdf:ID="John">
  <likes rdf:resource="#Apples" />
</Person>
...
<owl:Class rdf:ID="Proprietor">
  <equivalentClass rdf:resource="#Person"/>
</owl:Class>
```

Having two equivalent classes simply means that an instance of the first class is also an instance of the second class [6]. Thus, a new definition for atomic personal information in OWL can be established:

Personal information is any information where the triple in the ontology contains an instance of the Proprietor class.

Thus, recognizing personal information is simply a matter of recognizing triples that contain individuals of the proprietor class so that special considerations can be taken when performing operations. Additionally, declaring the proprietor class as an equivalent class to the person class ensures that non-proprietor entities are not mistakenly declared as individuals in the person class. For example, fictitious characters are not proprietors and thus cannot be in the same group of individuals who are proprietors (real identifiable persons).

4.2 Compound Statements

Compound personal statements require the following operations:

1. Privacy-reducing the compound personal statement to a set of atomic personal statements
The purpose of this step is to separate the personal information of different proprietors in order to enable the system to work with atomic statements only. This facilitates such processes as limiting each proprietor's access, if required, to his/her part in the compound personal information. For example: *Jane is the biological mother of Eddie* is reduced to *X is the biological mother of Eddie* and *Jane is the biological mother of Y*. X and Y can be represented by an instance of any class other than the proprietor class. However, *X and Y* must be uniquely declared (i.e., each must have its own URI) because they refer to different individuals. In the orphanage ontology we may have the following two pieces of personal information about Eddie:

(1) *The biological mother of Eddie is X*
(2) *The biological father is unknown*

Statement (1) reflects the fact that Eddie's mother is known to the orphanage, whereas his father is unknown. Eddie may be allowed to access this information but he would not be permitted to know X since Jane may not want Eddie to know her. The privacy-reducibility to atomic information gives many options for controlling access in different combinations, thus enhancing the semantics of the knowledge base. Privacy-reducibility ensures that each proprietor may access information about himself, but not the compound personal information, which contains the personal information of other proprietors. Similarly, Jane's information is obtained through accessing the atomic information: *Jane is the biological mother of Y.*

Once the compound personal statement has been reduced manually or automatically, each atomic statement may be freely exchanged, accessed or retained in the proprietor's domain, or with permission of the proprietor.

2. Declare the *Proprietor* class to be the equivalent class for the person class
This second step involves making the *Proprietor* class an equivalent class, and it is easily fulfilled because the proprietor in each atomic statement produced is the identifiable subject. For example in the atomic statement *The biological mother of Eddie is X,* Eddie is the proprietor.

3. Link the atomic statements produced using the construct owl:sameAs
In this step we link the atomic statements produced from the compound statement so they will not lose their meaning. If the original compound statement contains any additional properties linked to the person instances (individuals) in the statements, then these links are transferred to their atomic statements such that the proprietor retains all previous ontology links to the instance, and the unknown (in the context of atomic statement), such as X or Y described previously, has no properties linked to it except the one appearing in the original compound statement. The unknown entity in

each statement is then cross-linked with the individuals they represent in the other atomic statements. This linking is accomplished with the OWL construct *owl:sameAs*.

Example. Consider the compound personal information *Eddie spoke with Jane* which is reduced to *Eddie spoke with X* and *Y spoke with Jane*. The X and Y here are variables that can be represented by an instance of any class other than the proprietor class. However, the X and Y variables in each statement must be uniquely declared (i.e., each must have its own URI) because they refer to different individuals. So the two atomic statements *Eddie spoke with X* and *Y spoke with Jane* have X declared the same as Jane and Y declared the same as Eddie, as illustrated by figure 2.

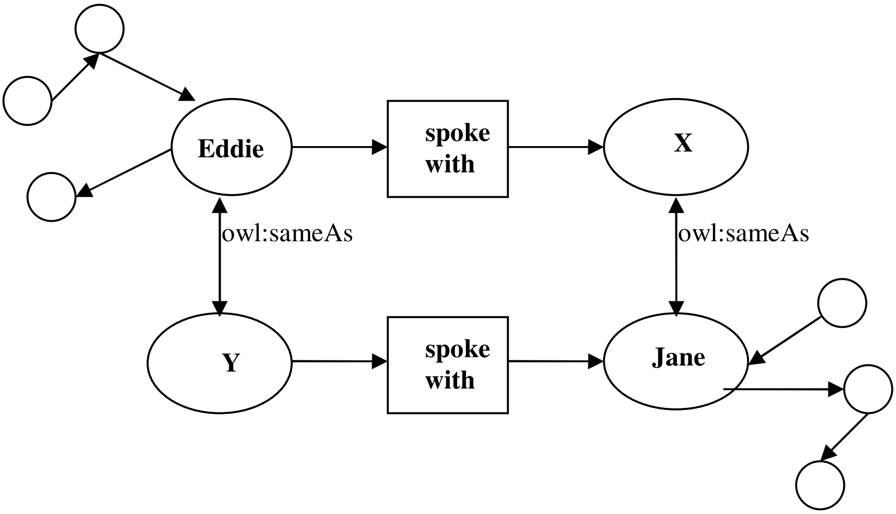


Fig. 2. Atomic Personal Information in OWL

The privacy-reduced ontology could have the following OWL syntax:

```
<owl:Class rdf:ID="Person"/>
<owl:Class rdf:ID="Proprietor">
  <equivalentClass rdf:resource="#Person"/>
</owl:Class>
...
```

These declarations are in Eddie's domain:

```
<owl:ObjectProperty rdf:ID="spokeWith"/>
<Person rdf:ID="Eddie">
  <spokeWith rdf:resource="#X" />
  <owl:sameAs rdf:resource="#Y" />
</Person>
...
```

These declarations are in Jane's domain:

```
<owl:ObjectProperty rdf:ID="spokeWith"/>
<Person rdf:ID="Jane">
  <spokeWith rdf:resource="#Y" />
  <owl:sameAs rdf:resource="#X" />
</Person>
...
```

The *owl:sameAs* construct serves to unify the individual and *someone* in separate ontologies [14]. With our model, the system can distinguish between the personal information of each person. The same can be applied to compound personal statements containing more than two proprietors.

Example. *Jim and Harry like Mary* is reduced to three atomic statements:

Jim and Y like Z
X and Harry like Z
X and Y like Mary.

Each statement belongs to a single proprietor and shows only the proprietor's personal information. In each statement where the variables appear, X is declared the same as Jim, Y is declared the same as Harry, and Z is declared the same as Mary.

In summary, representing personal information in an OWL ontology requires the following:

(1) The *Proprietor* class is declared to be an equivalent class for any class whose individuals are single, identifiable humans.
(2) Compound private statements are reduced to atomic private statements.
(3) The *owl:sameAs* construct is used to link privacy-reduced statements.

In addition to these requirements, we introduce two restrictions to OWL.

5 Restrictions

Although the OWL specifications allow class hierarchies to be created, for the purpose of modeling personal information our model requires that any instance (individual) of a person class that refers to a single identifiable human (instance of the class proprietor) can have no subclasses. This restriction is based simply on the fact that a person in our sense represents an identity that cannot logically be considered a group of entities and cannot have sub-hierarchies. Although this may seem to be a severe restriction, this requirement is necessary. For example, one might consider John to be a collection of the body parts that make him, and we can model this information by making all his body parts subclasses of John. A better way to represent this information would be to make a class of body parts, have all the body parts be instances (individuals) of that class, then relate them to John with a property such as the *PartOf* property. The rationale for this is simple: subclasses can inherit the properties of the superclass. Thus, an identifiable person must be represented as an instance of a class (individual) because a person cannot be a class.

We also require not using the constructs *owl:differentFrom* and *owl:allDifferent* in the personal information model introduced here because it would create a conflict with the *owl:sameAs* construct, which we use to link the variable instances (individuals, e.g., X, Y, Z in previous example) with the person instances to which they refer.

In summary, we require two restrictions to be applied when modeling personal information in OWL:

(1) An identifiable person must be represented as an instance of a class.
(2) *owl:differentFrom* and *owl:allDifferent* may not be used on proprietors.

6 Personal Information Closure

We have introduced the ontology to model compound and atomic personal statements in OWL. In OWL, a personal information triple consists of an instance (individual) of a person, of a property, and of any other instance to complete the triple. It does not matter if the person is the subject or the object value. For example, *a disease infected Peter* is a personal information triple as is the personal information triple *Peter owns a yacht*. In this sense, we can think of personal information as the circle of all the properties and individuals linked to a person within a distance of one, where the person (proprietor) is at the center. Properties in OWL may have no value (target individual), such as the property *fell* in *Peter fell*, so the pair of the person the property links to it is also personal information. In this sense, we can think of personal information as the circle of all the properties and instances surrounding an individual of class proprietor. Thus, we introduce the concept of personal information closure. The personal information closure demarcates the extent of personal information centered on the proprietor and includes all personal information triples.

Example. Figure 3 illustrates the personal information closure inside the dashed line.

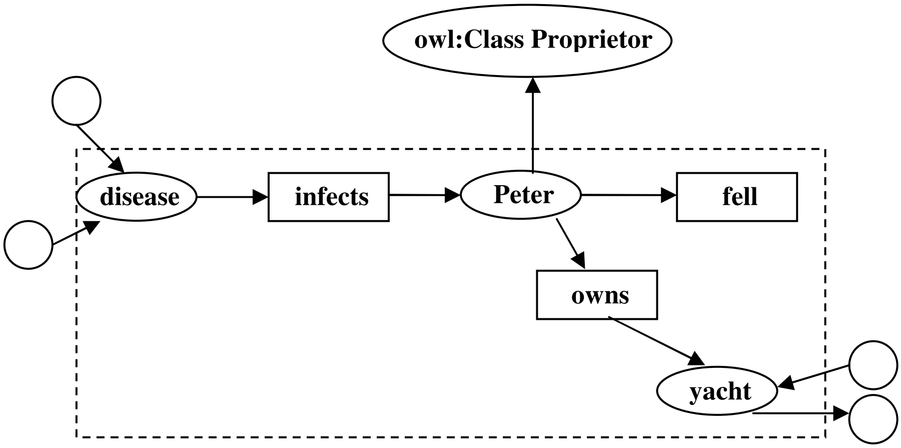


Fig. 3. The Personal Information Closure

Some OWL constructs may expand the scope of the personal information closure beyond the triple containing the proprietor. When a property is declared to be transitive using *owl:transitiveProperty*, and that property has an individual of type proprietor as a source or a target, then the scope of personal information goes beyond the triple or pair described earlier.

Example. Consider the information that *Richard owns the company and the company owns the agency and the agency owns a vehicle.* It can have the property *owns* declared to be transitive. This means that *Richard owns the company* and *Richard owns the agency* and *Richard owns the vehicle*, and all three statements are personal information. Thus, when the property is transitive and has an individual of type proprietor as a source or a target, then the chain of all instances linked by the same property to the person instance is personal information, as shown in figure 4.

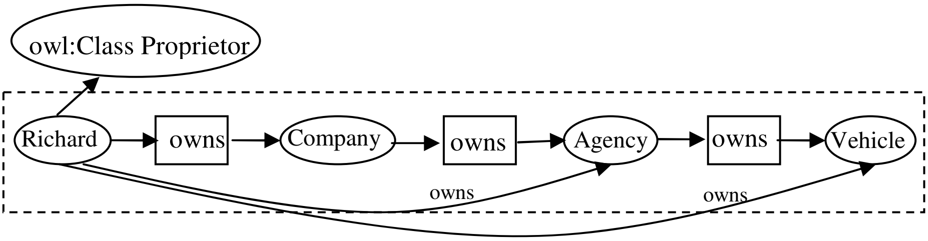


Fig. 4. The transitivity construct in OWL and personal information closure

The ontology in Figure 4 corresponds to the following OWL declarations:

```
<owl:Class rdf:ID="Person"/>
<owl:Class rdf:ID="Proprietor">
  <equivalentClass rdf:resource="#Person"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="owns">
  <rdf:type rdf:resource="&owl;TransitiveProperty" />
</owl:ObjectProperty>
...
<Person rdf:ID="Richard">
  <owns rdf:resource="#Company" />
</Person >
<owl:Thing rdf:ID="Company">
  <owns rdf:resource="#Agency" />
</owl:Thing>
<owl:Thing rdf:ID="Agency">
  < owns rdf:resource="#Vehicle" />
</owl:Thing>
...
```

Personal information includes the chain of instances linked to the person. In this transitive chain, other embedded statements, such as *The company owns the agency* and *the company owns the vehicle* are not personal information. The same applies if the person is the target of the transitive property as in *The vehicle belongs to the agency and the agency belongs to the company and the company belongs to Richard.* The embedded information that has the person as its target, such as *the vehicle belongs to Richard* and the agency belongs *to Richard*, is personal information.

The transitivity of properties has expanded the scope of personal information beyond the triple described earlier. The personal information in the case of a transitive chain is *implied* personal information.

Definition. *Personal Information Closure* is the set of all entities that consist of sets of implied personal information triples.

The OWL language allows creation of property hierarchies by declaring one or more properties to be *subproperties* of other properties. For example, in *Charles painted his house*, property *painted* can be declared as a subproperty of the *renovated* property. A reasoner can deduce that since *Charles* painted his house, then he has done some renovation to his house. For our personal information model, whenever the *owl:subPropertyOf* construct is used on a property linked to an individual of type proprietor, then (a) the pair of the person and the subproperty or superproperty, (b) the triple of the person, and (c) the property and the value or object on the other side of the property are included in the personal information closure, as illustrated in figure 5.

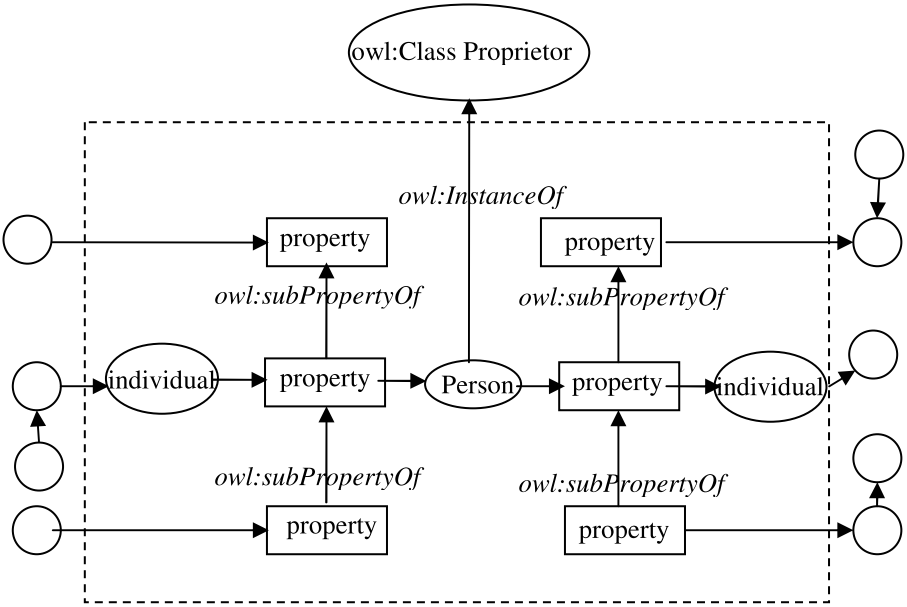


Fig. 5. Property hierarchies in OWL and Personal Information Closure

OWL also allows properties to be declared equivalent properties that relate individuals to the same set of other individuals. For example, if X is related to Y by property A, and B is declared equivalent to A, then X is also related to Y by B property. The difference between equivalent properties and subproperties is that equivalent properties are subproperties of each other. So if A is equivalent to B, then A is a subproperty of B and B is a subproperty of A. However, if we declared A to be a subproperty of B, then we cannot say that B is a subproperty of A. In practice, using the subproperty construct relates the same set of individuals with different properties; however, the individual will not be related to the individuals of other properties in the hierarchy. Using the equivalent property relates the individual with all the individuals related to the equivalent properties. So, for our personal information model, whenever the *owl:EquivalentProperty* construct is used on a new property to equal it with a property linked to an instance of a person, then the personal information closure includes all the individuals related to the original property, plus all the individuals related to the equivalent property, as illustrated in figure 6.

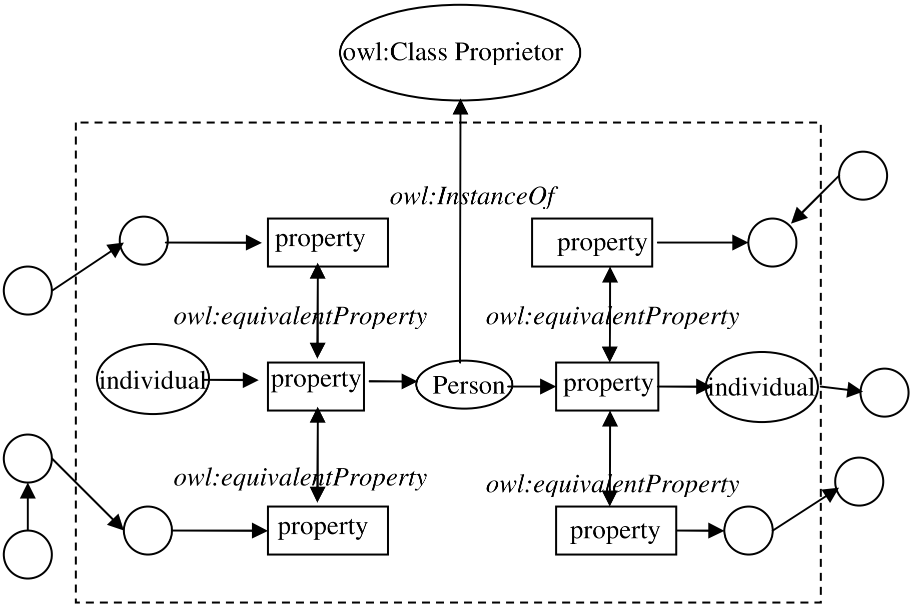


Fig. 6. Property equivalence in OWL and Personal Information Closure

7 Conclusion

We have introduced a scheme to model personal information in OWL including identification of personal information, so that the system can handle units of personal information. The proposed model introduces personal information ontology below the levels of policy and privacy preferences. We also investigated the effects of the various OWL constructs through introducing a set of requirements and restrictions when dealing with personal information. We have also introduced the notion of personal information closure, which defines the sphere of the proprietor's personal information, and investigated the notions of transitivity when property equivalence and hierarchies are used. OWL has a good potential for being the tool for implementing personal information, especially because of its expressive power in facilitating links between atomic statements that result from privacy-reducibility.

Clearly the methodology represents a new direction toward embedding privacy in the semantic web. Subsequent research in this area will introduce more precise specification of this task and may further investigate the rules of possession, exchange, retention, disclosure, sharing, and other operations.

References

1. Al-Fedaghi, S.: Crossing Privacy, Information, and Ethics. 17th International Conference, Information Resources Management Association (IRMA 2006), Washington, DC, USA. 21-24 May 2006.
2. Al-Fedaghi, S.: How Would Aristotle Define Privacy?. The First International Conference on Legal, Security and Privacy Issues in IT, Hamburg, Germany. 30 April – 3 May 2006.
3. Al-Fedaghi, S.: How to Calculate the Information Privacy. Proceedings of the Third Annual Conference on Privacy, Security and Trust, St. Andrews, New Brunswick, Canada. 12-14 Oct. 2005. <http://www.lib.unb.ca/Texts/PST/2005/pdf/ fedaghi.pdf>
4. Al-Fedaghi, S.: The 'Right to Be Let Alone' and Personal Information, Proceedings of the 7th International Conference on Enterprise Information Systems, Miami, USA. 2005.
5. Antoniou, G., van Harmelen, F.: Web Ontology Language: OWL. In: S. Staab & R. Studer (eds.): Handbook on Ontologies in Information Systems. Springer-Verlag (2004) 67-92.
6. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., Stein, L. A.: Owl web ontology language reference. W3C, 10 Feb. 2004. <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>
7. Brickley, D. (ed.).: Resource Description Framework (RDF) Schema Specification 1.0. W3C, 27 March 2000, <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>.
8. Gleick, J.: Behind Closed Doors; Big Brother Is Us. New York Times, 29 Sept. 1996.
9. Heflin, J.: OWL Web Ontology Language Use Cases and Requirements. W3C. 2004. <http://www.w3.org/TR/webont-req/>.
10. Horrocks, I., Patel-Schneider, P. F., van Harmelen, F.: From SHIQ and RDF to OWL: The Making of a Web Ontology Language. J. of Web Semantics. Vol. 1, 2003. 7-26.
11. Kang, J.: Information Privacy in Cyberspace Transactions. 50 Stanford Law Review 1193, 1212-20, Apr. 1998.
12. Kim, A., Hoffman, L. J., Martin, C. Dianne.: Building Privacy into the Semantic Web: An Ontology Needed Now. Proc. of Semantic Web Workshop, Hawaii, USA, 2002.
13. Perri, 6.: The Future of Privacy, Volume 1: Private life and Public Policy. Demos, London, 1998.
14. Smith, M. K., Welty, C., McGuinness, D.: OWL Web Ontology Language Guide. W3C. 2004. <http://www.w3.org/TR/2004/REC-owl-guide-20040210/> .
15. Stoddart, J. Annual Report to Parliament 2004-2005, Office of the Privacy Commissioner of Canada.
16. Wacks, R.: Privacy in Cyperspace. Privacy and Loyalty. Ed. P. Birks. New York: Clarendon Press, 1997. New York. 91-112.

A Semantic Reputation Mechanism in P2P Semantic Web

Wei Wang, Guosun Zeng, and Lulai Yuan

Department of Computer Science and Technology, Tongji University,
Shanghai 201804, China
Tongji Branch, National Engineering & Technology Center of
High Performance Computer, Shanghai 201804, China
willtongji@gmail.com

Abstract. Regarding the uncensored nature of the Semantic Web, the question of how much credence to give each information source is a main problem. We cannot expect each user to know the trustworthiness of each source. We tackle this problem by employing a semantic reputation mechanism which enables P2P Semantic Web to utilize reputation mechanism based on semantic similarity among peers. Our experiments show that the system with this mechanism outperforms the system without it. We hope that this method will help move the Semantic Web closer to fulfilling its promise.

1 Introduction

The goal of Semantic Web is to build a web of meaning. Semantic Web will consist of a distributed environment of shared and interoperable ontologies, which have emerged as common formalisms for knowledge representation, and anyone can be an information producer or consume of others' information. So, it is promising to combine the P2P network with Semantic Web technology. On one hand, P2P networks can help semantic web with sharing knowledge; on the other hand, P2P networks use the semantic concept to query routing and efficient content-based location. However, even after these are in wide use, we still need to address the major issue of how to decide how trustworthy each information source is. In order to use it well, the Semantic Web should be developed in a trustworthy environment. Our method is to introduce a semantic reputation mechanism which is based on peers' semantic similarity in P2P Semantic Web to solve the of lack trust.

The rest of this paper is organized as follows. We review some related work in Section 2. Section 3 introduces semantic reputation mechanism. We first describe the overview of the system model, and then propose the trust evaluation algorithm to compute the level of trust based on semantic similarity among peers. We evaluate our approach and analyze the experiments results in Section 4. Section 5 concludes the paper.

2 Related Work

Peer-to-peer networks are the sharing of computer resources and services by direct exchange between the systems. The features of P2P make it desirable to be an infrastructure for knowledge sharing in Semantic Web.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 682 – 688, 2006.

© Springer-Verlag Berlin Heidelberg 2006

So it is promising to use P2P infrastructure in Semantic Web. Zhuge et al. [1] propose a platform to support index-based path queries by incorporating a semantic overlay with an underlying structured P2P network. Arumugam et al. [2] propose P2P Semantic Web (PSW) which is a platform for creating, exchanging and sharing knowledge so that we can obtain more useful information. Ernst [3] uses P2P to support the semantic web to link semantic definitions. Broekstra et al. [4] build SWAP to combine Semantic Web with P2P.

On the other hand, reputation-based trust management [5] has been identified as a promising solution to the problem of trust in P2P networks. The main goal of the reputation mechanism is to take the reputation information that is locally generated as a result of an interaction between peers, and spread it throughout the network to produce a global reputation. Various reputation management mechanisms have been developed. Gupta et al. [6] present a partially centralized mechanism using reputation mechanisms. Kamvar et al. [7] propose the EigenTrust algorithm, which produces global reputation ratings for users based on their history behaviors. Despotovic et al. [8] document the P2P reputation techniques particularly well.

In this paper, we propose a semantic reputation mechanism to utilize reputation in Semantic Web. This mechanism can take the advantage of both P2P infrastructure and reputation system.

3 A Semantic Reputation Mechanism

3.1 Overview of the Architecture Model

Based on the past research on knowledge sharing in a multi-ontology environment and reputation mechanism in P2P networks, our approach may be implemented for any unstructured P2P network.

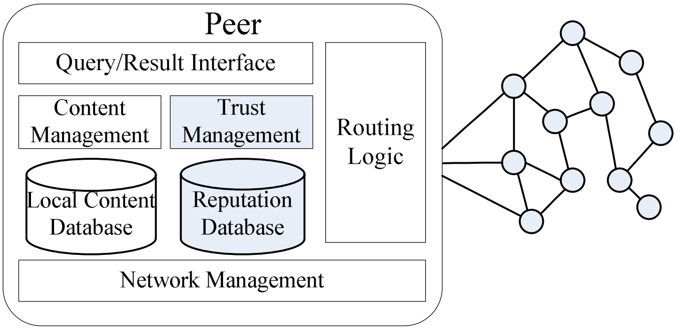


Fig. 1. Overview of the Architecture Model

Figure 1 shows the basic building blocks of our architecture. We assume that each peer provides a unique peer identifier. Similar to file sharing networks each peer may publish all resources from its local content database, and other peers can discover them by their requests [9]. All information is wrapped as RDF statements and stored in an RDF repository.

Another main component is the trust management with the support of the reputation database which stores reputation data of the peers. The main goal of the reputation mechanism is to take the reputation information that is locally generated as a result of an interaction between peers, and spread it throughout the network to produce a global reputation rating for the network nodes.

3.2 Reputation Mechanism

We introduce a reputation model offering a viable solution to encouraging trustworthy behavior in Semantic Web. The key presumptions are that the participants of an online community engage in repeated interactions and that the information about their past doings is indicative of their future performance and as such will influence it.

Our idea is to find an important feature of trust within P2P semantic web systems, that is the successful cooperation probability between two peers, and to try to estimate it using Bayesian method, as the Bayesian method supports a statistical evidence for trust analysis.

For the sake of simplicity, we only consider a system within the same context during a period of time. For two peers x and y, the successful cooperation probability between them is denoted by θ. There may have direct interactions between them, there may also have other intermediate entities and each of them has direct experiences with x and y. On the one hand, if there are direct interactions between x and y, we can obtain direct probability of successful cooperation, which is called *local reputation* value, and denoted by θ_{lr} . If there is an intermediate peer z between x and y, z and y, then, we can also obtain an indirect probability of successful cooperation between x and y, which is called *recommendation reputation* value, and denoted by θ_{rr} . So, there are two kinds of probabilities of successful cooperation. We will combine these two kinds of probabilities to be the estimator of successful cooperation probability. The whole process can be seen in Figure 2.

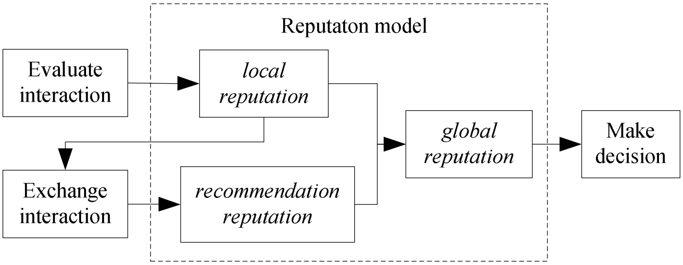


Fig.2. Overview of the reputation model

For the interaction probability, here we use Bayesian approach to compute its estimator. Suppose that the probability of successful cooperation between two peers is modeled with a Beta prior distribution, which is used to represent probability distribution of binary events [13]. Using Bayesian method, we get the Bayesian estimator of the probability, which is,

$$\hat{\theta}_{lr} = E(Beta(\theta \mid u+1, v+1)) = \frac{u+1}{u+v+2} \tag{1}$$

With respect to recommendation probability, we use the following formula to be its estimator:

$$\hat{\theta}_{rr} = E(Beta(\theta \mid u_1 + u_2 + 1, v_1 + v_2 + 1)) = \frac{u_1 + u_2 + 1}{n_1 + n_2 + 2} \tag{2}$$

in which, n_1 (n_2)is the number of interaction between *x* and *z* (*z* and *y*), and u_1 (u_2) is the number of successful cooperation. Then *global reputation value* can be expressed by the formula:

$$\hat{\theta} = \lambda\, \hat{\theta}_{dr} + (1\text{-}\lambda)\, \frac{\sum s \cdot (u+1)}{\sum s \cdot (u+v) + 2} \tag{3}$$

where *λ* is the weight to represent the importance of these two probabilities and is decided by the personal characteristics of the peers; *s* is the semantic similarity which defied below.

The problem of the trust management based on the peers' reputations can now be stated simply as follows: define a strategy to aggregate the available feedback and output an estimate of the trustworthiness of any given peer so that trustworthy behavior of the peers is encouraged. Different from the other reputation model, we use semantic similarity between peers to aggregate the feedback.

3.3 Semantic Similarity-Based Aggregation

Recent studies [12] have provided empirical evidence that users tend to rely upon recommendations from friends and family members, i.e., people they trust, more than upon those from online systems. We believe that given an application domain, such as, for instance, the book-reading domain, people's trusted peers are considerably more similar to their sources of trust than arbitrary peers. More formally, let *A* denote the set of all community members and trust(*x*) the set of all users trusted by *x*: [12]

$$\forall x \in A : \frac{\sum_{y \in trust(x)} sim(x, y)}{|trust(x)|} \gg \frac{\sum_{z \in A \setminus trust(x)} sim(x, z)}{|A \setminus trust(x)|} \tag{4}$$

For instance, given that peer *x* is interested in Sci-Fi and AI, chances that *y*, trusted by *x*, also likes these two topics are much higher than for peer *z* not explicitly trusted by *x*. Various social processes are involved, such as participation in those social groups that best reflect our own interests and desires.

In light of this, in a file-sharing P2P network a single document (or the content of document) can be classified into at least one topic. So, the semantic reputation model measures the similarity between peers. We use *P*= {<*Ti*, *λi*>, *i*=1, 2... *m*} to describe the participant topic, where *Ti* denotes a peer's topic, and *λi* shows the degree of interest to *Ti*. The similarity between two different peers is described as the similarity among the sets of topics. In case the peers in the network share a common topic hierarchy our aggregate algorithm exploits the semantic similarity between peers. The study of semantic similarity between lexically expressed concepts has been a part of natural language processing for many years. Based on the method in [10] and [11], the following equation is proposed to measure the similarity between two peers:

$$Sim(P_1, P_2) = \sum_{j=1}^{|P_2|} \sum_{i=1}^{|P|_1} [Sim(T_i, T_j) \times (\lambda_i \times \lambda_j)] \tag{5}$$

Here, |*P*1| and |*P*2| are the topic numbers in the two peers. The similarity between the sets of topics of each other is calculated by summing up products of the similarity value between two topics separately selected from *P*1, *P*2. More detail can be found in [10]. As have mentioned above, the more similar the two peers are the great their established trust would be considered. So, given a similarity threshold, we can compute the peer's trust value according Equation 3.

4 Simulation Results and Evaluation

We evaluate our approach in a simulation of a content sharing system in a peer-to-peer network from original Gnutella-like network. Every peer only knows other peers directly connected with it and a few content providers at the beginning.

Every peer has a topic vector. The topic is composed of five elements: music, movie, image, document and software. The value of each element indicates the strength of the peer's topic in the corresponding content type. Every peer keeps two lists. One is the peer list that records all the other peers that the peer has interacted with and its trust values in these peers. The other is the content provider list that records the known content providers and the corresponding reputation data representing the peer's trusts in these content providers. Each content provider has a capability vector showing its capabilities in different aspects, i.e. providing content with different types, qualities and download speeds. Our experiments involve 10 different content providers and 400 peers. λ in formula 3 is set to 0.8.

The goal of the experiment is to see if a reputation mechanism helps peers to select content providers that match better their preferences. Therefore we first compare the performance (in terms of percentage of successful recommendations) of a system consisting of peers with reputation mechanism and a system consisting of peers that represent normal mechanism. Successful recommendations are those positive recommendations when peers are satisfied with interactions with content providers with good reputation. If a peer gets a negative recommendation of a content provider, it will not interact with the content provider.

Figure 3 shows that the system using reputation mechanism performs better than the system without reputation mechanism, especially when the number of interactions is large. This is profitable for the large, uncensored Semantic Web. In some sense, a peer's trust networks can be viewed as the model of a specified content provider from the peer's personal perspective. We also notice that there is a minimum at about 220 interactions. This is because, at the initial state, the trust relationships between peers have not been constructed completely, so the successful recommendation decreases. After some interaction, the trust relationship is built, and the whole system improves exponentially.

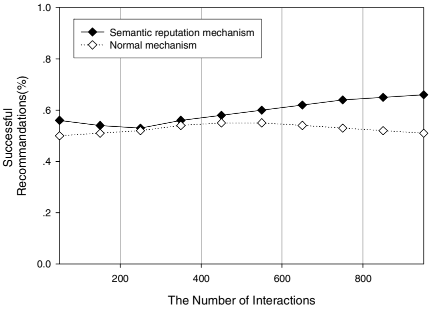


Fig. 3. Semantic reputation mechanism vs. normal mechanism

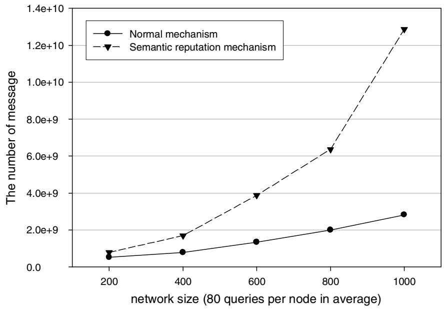


Fig. 4. A Performance with varying network size

Then we evaluate the querying performance of our approach. Figure 4 shows the number of messages transmitted in the P2P semantic web system with increasing network size. By applying our mechanism, the number of messages only increases linearly with the network size compared to the exponential-like increase of normal mechanism, which shows great scalability. Trust can help to reduce the load on the P2P network by leading interaction to the most highly reputed peers in a given matter, avoiding polling untrusted peers.

In the real web semantic, the model of content providers might be more complex and required the use of a more complex semantic-based mechanism. If we build a more complex reputation mechanism and add more aspects into it, the system performance might be improved.

5 Conclusions

In this paper, we propose a semantic-based reputation system to solve the of lack trust in Semantic Web. We evaluated our approach in a simulation of a content sharing system in a P2P Semantic Web. Our experiments show that the system with reputation mechanism outperforms the system without it. Applying this approach to a real P2P Semantic Web for computational services is particular promising.

Acknowledgements

This research was partially supported by the National Natural Science Foundation of China under grant of 60173026, the Ministry of Education key project under grant of 105071 and SEC E-Institute: Shanghai High Institutions Grid under grant of 200301.

References

1. Zhuge H., Sun X., et al.: A Scalable P2P Platform for the Knowledge Grid. IEEE Transactions on Knowledge and Data Engineering, 17(12) (2005)1721–1736
2. Arumugam, M., Sheth, A., Arpinar, I. B.: Towards Peer-to-Peer Semantic Web: A Distributed Environment for Sharing Semantic Knowledge on the Web. Technical report, Large Scale Distributed Information Systems Lab, University of Georgia, (2001)
3. Ernst, J.: Peer-to-Peer infrastructure supporting the semantic web. Int. Semantic Web Symposium, Stanford Univ. (2001)
4. Haase, P., Broekstra, J., Ehrig, M., et al.: Bibster-a semantics-based bibliographic peer-to-peer system. In Proceedings of the International Semantic Web Conference (ISWC), (2004)
5. Resnick, P., Zeckhauser, R., Friedman, R., et al.: Reputation systems. Communications of the ACM, 43(12) (2000) 45–48
6. Gupta, M., Judge, P., Ammar, M.: A reputation system for peer-to-peer networks. In Proceedings of the NOSSDAV'03 Conference, Monterey, CA, (2003)
7. Kamvar, S. D., Schlosser, M. T., GARCIA-MOLINA, H.: The EigenTrust algorithm for reputation management in P2P networks. In Proceedings of the 12th International Conference on World Wide Web, ACM Press, (2003) 640–651
8. Despotovic, Z., Aberer, K.: P2P reputation management: Probabilistic estimation vs. social networks. Computer Networks 50 (2006) 485–500
9. Loser, A., Tempich, C., Quilitz, B., et al.: Searching dynamic communities with personal indexes. In 3rd. International Semantic Web Conference (ISWC) Galway, (2005)
10. Chen H.H., Jin, H., et al.: SemreX: A Semantic Similarity Based P2P Overlay Network. Journal of Software, 17(5) (2006) 1170–1181
11. Li, Y., Bandar, Z, McLean, D.: An Approach for measuring semantic similarity between words using semantic multiple information sources. IEEE Transactions on Knowledge and Data Engineering, 15 (2003)
12. Ziegler, C.N., Lausen, G.: Analyzing Correlation between Trust and User Similarity in Online Communities. In Proceedings of the 2nd International Conference on Trust Management, (2004)
13. Heckerman D.: A tutorial on learning with Bayesian networks. Technical Report, MSR-TR-95-06, Microsoft Research Advanced Technology Division, Microsoft Corporation (1995)

Client and Server Anonymity Preserving in P2P Networks

Byungryong Kim[1]

[1] DongBang Data Technonogy Co., Ltd. No.417, Hanshin IT Tower #235, Kuro-Dong,
Kuro-Ku, Seoul, Korea, 152-050
doolyn@gmail.com

Abstract. The participating nodes exchange information without knowing who is the original sender in P2P networks of basic form. Packets are relayed through the adjacent nodes and do not contain identity information about the sender. Since these packets are passed through a dynamically-formed path and since the final destination is not known until the last time, it is impossible to know who has sent it in the beginning and who will be the final recipient. The anonymity, however, breaks down at download/upload time because the IP address of the host from which the data is downloaded can be known to the outside. We propose a technique to provide anonymity for both the client and the server node in unstructured/structured P2P network. A random node along the path between the client and the server node is selected as an agent node and works as a proxy: the client will see it as the server and the server looks at it as the client, hence protecting the identity of the client and the server from each other.

1 Introduction

Peer-to-Peer(P2P) file sharing is now very popular and comes into the spotlight as new application in internet environment. Many techniques for P2P file sharing are currently being invented. P2P file sharing is classified into unstructured p2p system and structured p2p system. Unstructured p2p system includes Freenet[1], Gnutella[2] and structured p2p system includes Chord[3], Tapestry[4], CAN[5]. Many users are using file-sharing software by means of p2p application at this moment. Both, however, do not provide anonymity for server locations and could expose servers to DoS or storage flooding attack or anonymity-breaking attacks[6,7,8,9].

Many users are using file-sharing software by means of p2p application at this moment and many of them are violating copyright while performing p2p file sharing as well. While there are users who use such programs knowing that they are violating copyright, most of users use file-sharing programs without knowing it.

Downloading content with copyright means that the content is downloaded by somebody else at the same time and this content exists in my computer. At the end users having this content may have to be responsible in any way. In p2p system, where host itself shall be responsible, the exposure of identity may cause malicious attack. Therefore many p2p users want to conceal that they are performing file sharing and Freenet is an example of systems formed for this trend.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 689 – 695, 2006.
© Springer-Verlag Berlin Heidelberg 2006

This study proposes technique to protect identity of client and server, problem raised above. Proposed technique is designed to secure the anonymity of server and client in Gnutella and Chord. An intermediate node is selected as the agent who goes between the client and the server. This agent will pose itself as the server to the client and creates this illusion by replacing the true server IP with its own one in the query hit message packets. It also relays the client's content request to the true server and relays the data back to the client pretending as the true server. In structured p2p system, proposed technique is designed to secure the anonymity of server and client in Chord, a representative example of structured p2p based on distributed hash table. The problem of load balancing, which may be found, is effectively achieved as well. At Chord each node manages neighbor-set, the closest successor group. When receiving retrieval request, random node of neighbor-set performs the role of relay server between server and client. It was made such that request of this relay server is not concentrated on a certain successor.

This study is composed of four parts: chapter 2 summarizes previous researches on providing anonymity in P2P network; chapter 3 will look into mutual anonymity technique proposed in this study in detail; chapter 4 will discuss the proposed technique and make conclusions.

2 Related Researches

In a peer-to-peer system in which the identities of the participants are known, enforcing privacy is different from the traditional node anonymity problem. Some approaches use fixed servers or proxies to preserve the privacy. Publius[10] protects the identity of a publisher by distributing encrypted data and the k threshold key to a static, systemwide list of servers. However, in a peer-to-peer system, such a server list may not exist. APFS [11] has been proposed to achieve mutual anonymity in a peer-to-peer file sharing system. Some changes can be adopted so that it can be applied to streaming sessions. The primary goal for Freenet security is protecting the anonymity of requestors and inserters of files. As Freenet communication is not directed towards specific receivers, receiver anonymity is more accurately viewed as key anonymity, that is, hiding the key which is being requested or inserted. Anonymous point-to-point channels based on Chaum's mix-net scheme[12] have been implemented for email by the Mixmaster remailer[13] and for general TCP/IP traffic by onion routing[14]. Such channels are not in themselves easily suited to one-to-many publication, however, and are best viewed as a complement to Freenet since they do not provide file access and storage. Anonymity for consumers of information in the web context is provided by browser proxy services such as the Anonymizer[15], although they provide no protection for producers of information and do not protect consumers against logs kept by the services themselves. Private information retrieval schemes[16] provide much stronger guarantees for information consumers, but only to the extent of hiding which piece of information was retrieved from a particular server. In many cases, the fact of contacting a particular server in itself can reveal much about the information retrieved, which can only be counteracted by having every server hold all information. Reiter and Rubin's Crowds system[17] uses a similar method of proxing requests for consumers, although Crowds does not itself store information and does not protect information producers. Berthold *et al.* propose Web MIXes[18], a stronger system that uses message padding and reordering and dummy messages to increase security, but again does not protect information producers.

The Rewebber[19] provides a measure of anonymity for producers of web information by means of an encrypted URL service that is essentially the inverse of an anonymizing browser proxy, but has the same difficulty of providing no protection against the operator of the service itself. The Eternity proposal[20] seeks to archive information permanently and anonymously, although it lacks specifics on how to efficiently locate stored files, making it more akin to an anonymous backup service. Free Haven[21] is an interesting anonymous publication system that uses a trust network and file trading mechanism to provide greater server accountability while maintaining anonymity. MUTE[22] forces all intermediate nodes along the path between the client and the server node to work as proxies to protect the identities of the client and the server. Every node in the path including the client and the server thinks its previous node is the client and its next one the server. Therefore the data from the true server will be relayed node by node along the path causing a heavy traffic, especially for large multimedia files. Tarzan[23] is a peer-to-peer anonymous IP network overly. so it works with any internet application. Its peer-to-peer design makes it decentralized, scalable, and easy to manage. But Tarzan provides anonymity to either clients or servers. Mantis[24] is similar to Crowds in that there are helping nodes to propagate the request to the candidate servers anonymously.

3 Providing Anonymity Via Random Agent Nodes

In the proposed technique relay node is randomly selected and in the next session although the same server and client communicate each other relay node is selected. In our scheme, some agent nodes will be elected as the QueryHit packet traces back to Node 1. Suppose Node 6 and Node 3 are such agent nodes. Upon deciding to become an agent, Node 6 starts to modify the packet header: the IP Address and Port of Node 7 are replaced with those of Node 6. And the related information is saved in the SessionTable. Node 6 now acts as if it is the server who has sent the QueryHit packet. Node 3 also processes the packet similarly, but in this case, it thinks Node 6 is the server and sends the modified packet to Node 1 as a proxy for Node 6.

Node 1, upon receiving QueryHit packet, contacts Node 3, the first agent, and sends HTTP header to it requesting data. The UUID included in the packet, however, is that of Node 6. Node 3 knows that this packet should be delivered to Node 6 by noticing this mis-matching between UUID and the destination IP address. It builds a PUSH packet to request the indicated data from Node 6. Now Node 3 works as an agent between Node 1 and Node 6. The response from Node 6 will be relayed to Node 1 via Node 3. Similar events happen in Node 6 because it is another agent between Node 3 and the final server. Two agents, Node 3 and Node 6, will relay the data from the actual server to the client hiding the identities of both ends successfully

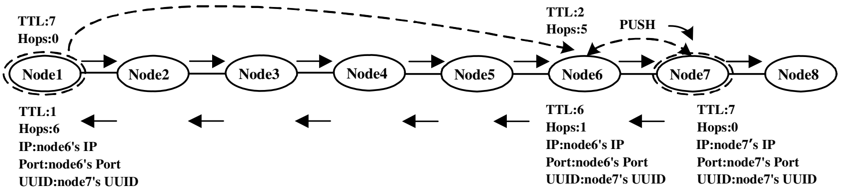


Fig. 1. Selection of agent nodes and flow of requested data through them

Secondly, Chord, one of P2P systems based on distributed hash table, enables very simple and effective retrieval. Identity protection technique by means of neighbor set second proposed in this study is based on Chord system.

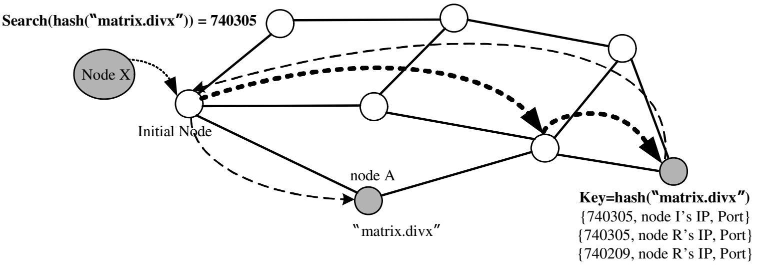

Search(hash("matrix.divx")) = 740305
Node X
Initial Node
node A
"matrix.divx"
Key=hash("matrix.divx")
{740305, node I's IP, Port}
{740305, node R's IP, Port}
{740209, node R's IP, Port}

Fig. 2. Download request flow in P2P system on distributed hash table base

Fig. 2 shows the normal retrieval in P2P system based on the most general distributed hash table. Node requesting retrieval searches key value from finger table of its own. Retrieval is performed by selecting the closest key within the range not exceeding the key value to be retrieved. Then key retrieval request is sent to node corresponding to retrieved key(initial node). In Fig. 1 node X was hashed to retrieve "matrix.divx" and hash value is 740305. Therefore this request is sent to the node closest to 740305 not exceeding 740305 among each entry of finger table owned by node X, and the same process is repeated at the node again as explained above. Finally if 740305 exists in P2P network, this request is sent to node 740305, if not the request is sent to the successor, the node closest to 740305 managing key 740305. The figure shows the case that correctly meeting node exists. Request arrives at the final destination, it finds key with 740305 among content list for which the node is responsible. If it does not exist the retrieval is failed and if it exists retrieved list is resent as shown on Fig. 2. This list includes information on IP address and Port having the requested contents. Recipient node X of the list requests contents download to node having wanted contents selected from the list.

If file transmission is performed as shown on Fig. 2, node A and node X, which are server, expose the identity. Accordingly packet can be intercepted by malicious node or node X and node A can be the target of attack.

Every node participating in Chord manages finger table. With this finger table intended contents can be quickly found. Therefore in order to maintain the finger table with latest condition getFinger message is periodically transmitted to successor. In proposed technique the closest successors, apart from the finger table, are managed as neighbor-set. If retrieval request on contents list of which responsibility is held by itself is received one node is randomly selected from neighbor-set. In addition the selected node deceives as if it has contents. Therefore client communicates with one node from the randomly selected neighbor-set. In Fig. 3 node 82 has "matrix.divx" file. hash("matrix.divx")= 45 and node 46, successor of 45 is responsible for key 45, so node 82 sends {45, {"matrix.divx", IP, port}}, key/value pair to node 46 which is responsible for key 45.

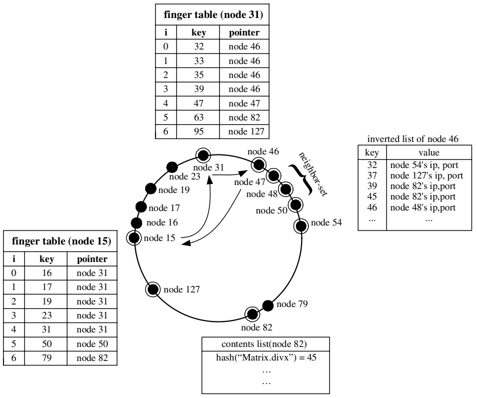


Fig. 3. Retrieval response flow by means of routing and neighbor-set in chord

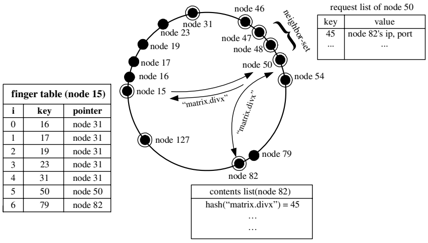


Fig. 4. Proxying by means of neighbor-set

Node 15 starts search(hash("matrix.divx")) request. According to routing of finger table this request is transmitted to node 46(i=3) again. Node 46 transmits key/value pair corresponding to key 45 from inverted list. It is normal routing. But in this study the result value is resent by means of neighbor-set. The method is as follows.

As shown on Fig. 3 every node manages neighbor-set, the closet successors' group to the node so node 46 manages 47, 48, 50 as neighbot-set. If retrieval request on inverted list for which node 46 is responsible is received, as shown on Fig. 4 one node is randomly selected from neighbor-set and ip and port information of selected node is changed into those of retrieved inverted list. For example as shown on Fig. 4 if randomly selected node is node 50, node 46 transmits {45,{node 50's IP, port}} to node 15 pretending as if it is retrieved. Node 46, before sending this, sends key/value pair of key 45, to node 50. Node 50 saves this to request list. Because node 15 received {45,{node 50's IP, port}}, contents download request is sent to node 50. Node 50 is able to know that 45 is smaller than its predecessor(node 48). Therefore if download request on smaller key than its predecessor is received, value to the key is found from the request list. If it is found contents download request is made to node corresponding value. If contents download is started, it is transmitted to initiator (node 15) as it is. Accordingly node 50 carries out the role of relay server between client server, node 15 and node 82. In this way node 15 and node 82 do not know who the server and client is. Because node 46 determines relay server randomly from neighbor-set when node 15 tries to download "matrix.divx" file saved at node 82, node(node 47, 48) other than node 50 is selected that it is hard to detect who the server and client is. Accordingly anonymity of server and client is secured.

In the proposed technique relay node is randomly selected as shown on Fig. 4 and in the next session although the same server and client communicate each other relay node is selected. So it is hard to correctly tell which node is server or client. In terms of load balancing as well, it is not fixed as static proxy that relay role is not concentrated on a specific node and the role is evenly dispersed.

4 Conclusions

Many P2P users are using file-sharing program. and users wish to conceal this fact. Finally both server and client want identity to be protected and by concealing the identity they wish to be protected from malicious attack. One is that We employed the idea of a proxy but not static one. The proxy is selected dynamically during the traverse of the QueryHit packet. Since the selection process is truly distributed, no one knows exactly how many proxies, or agents, are selected and where they are located. The agents are linked together only by neighbors: each agent knows only its previous and the succeeding one. We have designed the process such that only very limited number of agents are selected. Secondly, This paper proposed technique to be protected from malicious attack by protecting identity in the structured p2p system, chord. Each node manages successor list, neighbor-set and random node among the successors becomes proxy and provides file relay service between server and client. Node relaying file is randomly selected per session so it makes hard for attacker to know both the server and client at the same time. In the proposed technique it is safe from attack since mutual anonymity is secured by protecting identity of server and client.

References

1. I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, Freenet: A distributed anonymous information storage and retrieval system, In Workshop on Design Issues in Anonymity and Unobservability, pages 46.66, 2000., http://citeseer.nj.nec.com/clarke00freenet.html.
2. The Gnutella Protocol Specification v0.41 Document Revision 1.2., http://rfc-gnutella.sourceforge.net/developer/stable/index.html/
3. Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, Hari Balakrishnan, Chord: a scalable peer-to-peer lookup protocol for internet applications, IEEE/ACM Transactions on Networking (2003)
4. Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, and John Kubiatowicz, Tapestry: A Resilient Global-scale Overlay for Service Deployment, IEEE Journal on Selected Areas in Communications (2004)
5. Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, Scott Schenker, A scalable content-addressable network, Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications table of contents.
6. Neil Daswani, Hector Garcia-Molina, Query-flood DoS attacks in gnutella, Proceedings of the 9th ACM conference on Computer and communications security table of contents (2002)

7. P. Krishna Gummadi, Stefan Saroiu, Steven D. Gribble, A measurement study of Napster and Gnutella as examples of peer-to-peer file sharing systems, ACM SIGCOMM Computer Communication Review (2002)
8. A. Back, U. M¨oller, and A. Stiglic, Traffic analysis attacks and trade-offs in anonymity providing systems, In I. S. Moskowitz, editor, Information Hiding (IH 2001), pages 245.257. Springer-Verlag, LNCS 2137, 2001.
9. J. F. Raymond, Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems, In Workshop on Design Issues in Anonymity and Unobservability. Springer-Verlag, LNCS 2009, July 2000.
10. M. Waldman, A.D. Rubin, and L.F. Cranor, Publius: a robust, tamper-evident, censorship-resistant, web publishing system, in Proceedings of the Ninth USENIX Security Symposium, Denver, CO, USA (2000).
11. V. Scarlata, B. Levine, and C. Shields, "Responder anonymity and anonymous peer-to-peer file sharing," in Proc. of IEEE International Conference on Network Protocols (ICNP), Riverside, CA, 2001.
12. D.L. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, Communications of the ACM 24(2), 84-88 (1981).
13. L. Cottrell, Frequently asked questions about Mixmaster remailers, http://www.obscura.com/~loki/remailer/mixmaster-faq.html (2000).
14. Roger Dingledine, Nick Mathewson, Paul Syverson, Tor: The Second-Generation Onion Router, Proceedings of the 13th USENIX Security Symposium (2004)
15. Anonymizer, http://www.anonymizer.com/ (2000).
16. B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, Private information retrieval, Journal of the ACM 45(6), 965-982 (1998).
17. M.K. Reiter and A.D. Rubin, Anonymous web transactions with Crowds, Communications of the ACM 42(2), 32-38 (1999).
18. O. Berthold, H. Federrath, and S. Kopsell, Web MIXes: a system for anonymous and unobservable Internet access, in Proceedings of the Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, USA. Springer: New York (2001).
19. The Rewebber, http://www.rewebber.de/ (2000).
20. R.J. Anderson, The Eternity service, in Proceedings of the 1st International Conference on the Theory and Applications of Cryptology (PRAGOCRYPT '96), Prague, Czech Republic (1996).
21. R. Dingledine, M.J. Freedman, and D. Molnar, The Free Haven project: distributed anonymous storage service, in Proceedings of the Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, USA. Springer: New York (2001).
22. MUTE: Simple, Anonymous File Sharing., http://mute-net.sourceforge.net/
23. Michael J. Freedman, Robert Morris, Tarzan: A Peer-to-Peer Anonymizing Network Layer, in Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), Cambridge, MA, USA (2002)
24. Stephen C. Bono, Christopher A. Soghoian, Fabian Monrose, Mantis: A Lightweight, Server-Anonymity Preserving, Searchable P2P, Information Security Institute of The Johns Hopkins University, Technical Report TR-2004-01-B-ISI-JHU (2004)

A Map Ontology Driven Approach to Natural Language Traffic Information Processing and Services

Hongwei Qi[1], Yuguang Liu[1], Huifeng Liu[1], Xiaowei Liu[1],
Yabo Wang[1], Toshikazu Fukushima[1], Yufei Zheng[2], Haitao Wang[2],
Qiangze Feng[2], Han Lu[2], Shi Wang[2], and Cungen Cao[2]

[1] NEC Laboratories, China,
11/F, Bldg.A, Innovation Plaza, Tsinghua Science Park, Beijing 100084, China
qihongwei@research.nec.com.cn

[2] Institute of Computing Technology,
Chinese Academy of Sciences, Beijing 100080, China
yfzheng@ict.ac.cn

Abstract. This paper proposes a map ontology driven approach to natural language traffic information processing, and also describes its evaluation results. Traffic congestion is considered a major urban problem whose solution has long been sought for by engineers and researchers. Recently, the idea of gathering traffic information from mobile users via short message service appears promising. However, the traffic information is difficult to process to achieve a high accuracy because of its direct, indirect and connotative expressions. The proposed map ontology consists of a set of concepts, attributes, relations and constraints on them. The map ontology plays two key roles: 1) a basis for natural language traffic information analysis, and 2) a basis for user query analysis. In this paper we present the major information processing modules and services for mobile users. Experimental results show that the proposed method can improve the traffic information processing accuracy to 93%–95%.

1 Introduction

Traffic congestion is considered one of the major urban problems whose solution has long been sought for by engineers, planners, and researchers[1]. Using traffic information to help alleviate congestion is a feasible solution in recent years. Due to advanced technologies nowadays, the solution is quite practical and economical.

Generally, the existing approach for gathering traffic information is typically achieved by fixed/mobile traffic sensors or other electronic devices[2,3]. However, it requires relatively large amounts of costly infrastructure and maintenance to cover wide areas and many roads. However, with a rapidly increasing number of mobile users (China has now over 400 million mobile users), an approach gathering traffic information by mobile users reporting via short message service

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 696–710, 2006.
© Springer-Verlag Berlin Heidelberg 2006

(SMS, natural language messaging)[4] appears promising. This approach will make traffic information gathering much easier and thus more practical.

However, traffic information reported via natural language messaging has its inherent geographic characteristics. It is primarily divided into three categories:

(1) Direct traffic information, in which the location of congestion is told directly, such as "West Bridge is jammed[1]", where "West Bridge" is the location of congestion (Fig. 1).

(2) Indirect traffic information, in which the location of congestion is instead told by a Point-Of-Interest geospatially near it, such as "Garden Park is slow traffic", where "Garden Bridge" is the actual location of congestion, and "Garden Park" is a Point-Of-Interest near the bridge (Fig. 1).

(3) Connotative traffic information, in which the location of congestion is instead told by a geospatially embodied one, such as "West Second Ring is slow traffic", where "West Second Ring" is a representative location, and "West Bridge", "Garden Bridge", "Gate Bridge" and "New Bridge" are actual locations of congestion (Fig. 1).

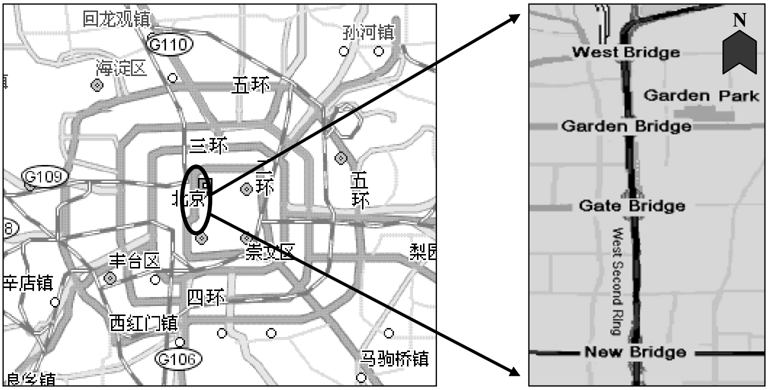


Fig. 1. A portion of Beijing City Map about West Second Ring

Generally, processing traffic information above is too difficult to achieve a high accuracy because of the inherent geographic characteristics above. Existing solutions including POETIC (POrtable Extendable Traffic Information Collator[5]) process police traffic reports by utilizing a geographic feature (such as road, bridge, etc.) name database, and a lexicon of around 1,100 words of traffic-status descriptions. POETIC can process direct traffic information with the database and the lexicon. However, it cannot process indirect and connotative traffic information.

In this paper, we present a new method of processing traffic information, especially the indirect and connotative categories, by traffic information processing

[1] Note that this paper processes Chinese traffic information, and in order to make examples international, all of them were translated into English (same below).

language (TIPL) based on a map ontology. The map ontology, constructed primarily in the spirit of Gruber's view of ontologies[6], provides geospatial knowledge of geographic features and traffic-status descriptions used to process traffic information, where geospatial knowledge comprises semantic definitions of geographic features and their geospatial relations. It also contains formal axioms for constraining the interpretation to those features and their relations. The map ontology can be described in OWL[7]. But currently, we described it in our own ontology language[8,9] in order to optimize the performance and efficiency of the traffic information service. TIPL, a high-level pseudo language, is designed as an inferencer to parse and analyze traffic information.

Using the processed traffic information above, we then provide a traffic information Query & Answer interface (called traffic information service system, TISS) to serve mobile users via SMS. Queries can be made in textual natural language, which is also processed by TIPL based on map ontology mentioned above.

The rest of the paper is organized as follows. Section 2 describes the architecture of TISS; Section 3 describes map ontology creation; Section 4 introduces traffic information processing; Section 5 introduces user query processing; Section 6 gives implementation and evaluation of TISS; Section 7 concludes the paper.

2 Architecture of TISS

TISS consists of three interacting modules (Fig. 2).

(1) Map ontology creation module

With the map ontology creator, this module translates geographic features and their geospatial relations in the electronic map into map ontology, and it also extracts traffic-status descriptions from historical traffic information and records them into the map ontology (See Sect. 3.3 for further information).

(2) Traffic information processing module

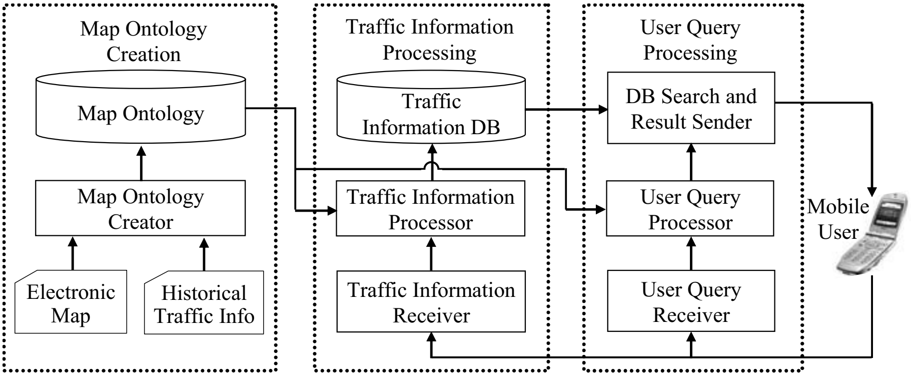
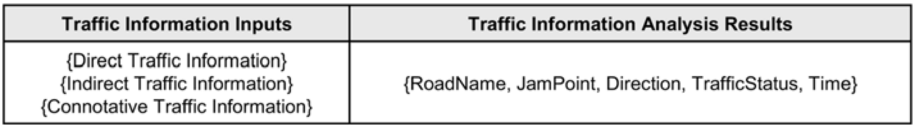


Fig. 2. Architecture of Traffic Information Service System (TISS)

As shown in Fig. 3, the input to the module, received by the traffic information receiver, is the direct, indirect or connotative traffic information reported by the mobile user. And the analysis results, stored in the traffic information DB (Fig. 4), are processed as terms {RoadName, JamPoint, Direction, TrafficStatus, Time}, where:

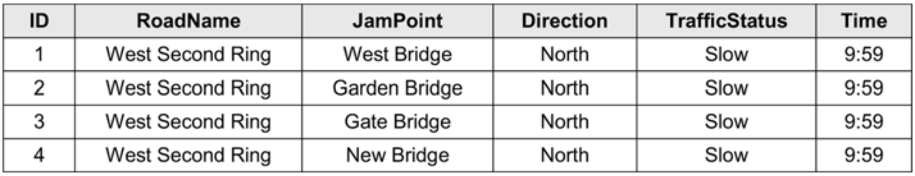
- RoadName: the name of the road where congestion happens;
- JamPoint: the representative name of the bridge, intersection, entrance, exit, Point-Of-Interest, etc., where congestion happens;
- Direction: the orientation of the congestion;
- TrafficStatus: the situation of the congestion, such as slow, jammed etc.;
- Time: the time when congestion happens.



Traffic Information Inputs	Traffic Information Analysis Results
{Direct Traffic Information} {Indirect Traffic Information} {Connotative Traffic Information}	{RoadName, JamPoint, Direction, TrafficStatus, Time}

Fig. 3. Inputs to & Analysis Results of Traffic Information Processor

The traffic information processor, programmed in TIPL and running based on map ontology, acts as an inferencer to translate the input into the analysis results in Fig. 3, that is to say, not only direct traffic info, but also indirect/connotative traffic info can be analyzed by the inferencer (See Sect. 4.2 for further information). As an example, for connotative traffic information input, “West Second Ring, north bound, is slow traffic”, the inferencer can produce the analysis results shown in Fig. 4.



ID	RoadName	JamPoint	Direction	TrafficStatus	Time
1	West Second Ring	West Bridge	North	Slow	9:59
2	West Second Ring	Garden Bridge	North	Slow	9:59
3	West Second Ring	Gate Bridge	North	Slow	9:59
4	West Second Ring	New Bridge	North	Slow	9:59

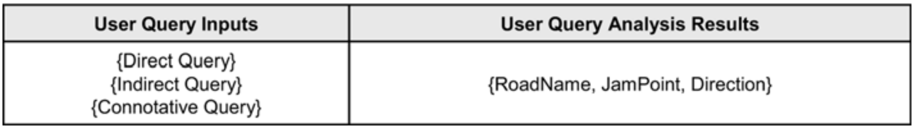
Fig. 4. A portion of Traffic Information DB

(3) User query processing module

Generally, traffic queries also have their inherent geographic characteristics. They are primarily divided into three categories:

1) Direct queries, in which the location is queried directly, such as “what about Garden Bridge?” (Fig. 1).

2) Indirect queries, in which the location is instead queried by a Point-Of-Interest geospatially near it, such as “what about Garden Park?”, where “Garden Bridge” is the location that should be checked in fact (Fig. 1).



User Query Inputs	User Query Analysis Results
{Direct Query} {Indirect Query} {Connotative Query}	{RoadName, JamPoint, Direction}

Fig. 5. Inputs to & Analysis Results of User Query Processor

3) Connotative queries, in which the location is queried with a geospatially embodied one, such as "what about West Second Ring?", where "West Bridge", "Garden Bridge", "Gate Bridge" and "New Bridge" are locations that should be checked in fact (Fig. 1).

So, as shown in Fig. 5, the input to the user query processing module, received by the user query receiver, is direct, indirect or connotative queries via SMS. And the analysis results are shown as terms RoadName, JamPoint, Direction, which are then organized as an SQL query to find an answer from the Traffic Information DB.

The user query processor, also programmed in TIPL and running based on map ontology, acts as an inferencer to translate the input into the analysis results in Fig. 5, that is to say, not only direct queries, but also indirect and connotative queries can be analyzed by the inferencer. As an example, for an indirect query input, "what about Garden Park?", the analysis result is like {West Second Ring, Garden Bridge, Null}, which is then organized as an SQL query like "select * from Traffic_information_DB where RoadName='West Second Ring' and JamPoint='Garden Bridge' ".

3 Map Ontology Creation

3.1 Background of Map Ontology

Different tasks of using geographic information have different requirements for such information. Meeting requirements of direct, indirect and connotative traffic information processing tasks demands an understanding of ontological aspects of geographic features.

Traditional map ontology usually aims to provide geographic knowledge, such as the name, latitude and longitude of geographic features[10]. However, semantic knowledge of geographic features is more important for our tasks. Such semantic knowledge includes:

(1) Attributes (semantic meaning) of the features. For example, "West Second Ring" is a "RoadName"; "Garden Bridge" is a "BridgeName". Here both "RoadName" and "BridgeName" are examples of attributes.

(2) Geospatial relations between these features. For example, "West Second Ring" is a segment of "Second Ring"; "Garden Bridge" is a point of "West Second Ring". Here both "segment of" and "point of" are examples of relations.

(3) Axioms to constrain the interpretation to these features and their relations. For example, if B (West Second Ring) is a segment of A (Second Ring), and at the same time, C (Garden Bridge) is a point of B (West Second Ring), then we can imply that C (Garden Bridge) is also a point of A (Second Ring).

Semantic knowledge is key to traffic information processing, especially for the indirect and connotative categories. Take the connotative traffic information "West Second Ring is in slow traffic." as an example. It can be inferred that since "West Second Ring" is "slow traffic", "West Bridge" ("Garden Bridge", "Gate Bridge" and "New Bridge") must also be "slow traffic" (see Fig. 1) based on the knowledge that "West Bridge" ("Garden Bridge", "Gate Bridge" and "New Bridge") is a point of "West Second Ring".

Therefore, integration of semantic knowledge of the geographic features into map ontology is key to processing direct, indirect or connotative traffic information with a high accuracy.

3.2 Preparing for Map Ontology Development

For the map ontology development, we obtained a copy of Beijing electronic map and a historical traffic information set, and a list of influential knowledge sources[11,12,13,14,15] recommended by experts in the transportation sector to make a comprehensive survey.

The results of the survey outline major definitions of map ontology mainly in the spirit of Gruber's view of ontologies[6]. The map knowledge space is composed of a set of concepts, which are interconnected with relations. In other words, each concept is described with attributes and its relations with the other concepts.

Definition 1. Formally, a concept C is a set of slot-value pairs. Slots are divided into two groups: attributes and conceptual relations. Therefore, we also say a concept is a set of attribute-value pairs and relation-concept pairs,

$$C = \text{def}\{<a_1, v_1>, <a_2, v_2>, \cdots, <a_i, v_i>\}\ U\ \{<r_1, C_1>, <r_2, C_2>, \cdots, <r_j, C_j>\}$$

where $a_1, a_2, \cdots, a_i$ are called the attributes of C and $v_1, v_2, \cdots, v_i$ are the values of these attributes. The attributes and their values represent the properties of the concept C. $C_1, C_2, \cdots, C_j$ are concepts. $r_1, r_2, \cdots, r_j$ are relations from C to $C_1, C_2, \cdots, C_j$.

In Definition 1, each attribute a_i or relation r_j defines a perspective of a concept, and several attributes and relations describe an integrated view of the concept. Fig. 6 is a schematic showing the meaning of the definition above.

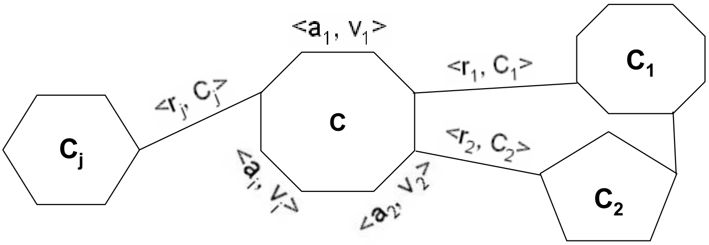


Fig. 6. Schematic of Concepts and their Relations

Take Fig. 1 for example. For the concept “Road”, its attribute-value pairs include “<RoadName, West Second Ring>”, “<Length, 4km>”, “<Direction, North-South>”, etc.; for the concept “Bridge”, its attribute-value pairs include “<BridgeName, West Bridge>”, “<Entrance, Port22>”, “<Exit, Port32>”, etc.. One relation example between the concepts “Road” and “Bridge” is “Point-of(x, y)”, such as “Point-of(West Bridge, West Second Ring)”.

3.3 Developing Map Ontology

Based on Definition 1, to develop map ontology, we consider two aspects: concepts and relations. In addition, axioms should also be considered in order to keep constraining the interpretation to these concepts and their relations. Axioms can also be used to make some inferences. To describe the three aspects above, we designed a frame-oriented map ontology (see Fig. 7). An attribute or relation may be associated with some facets for further constraining its interpretation and value(s). Below are some common facets:

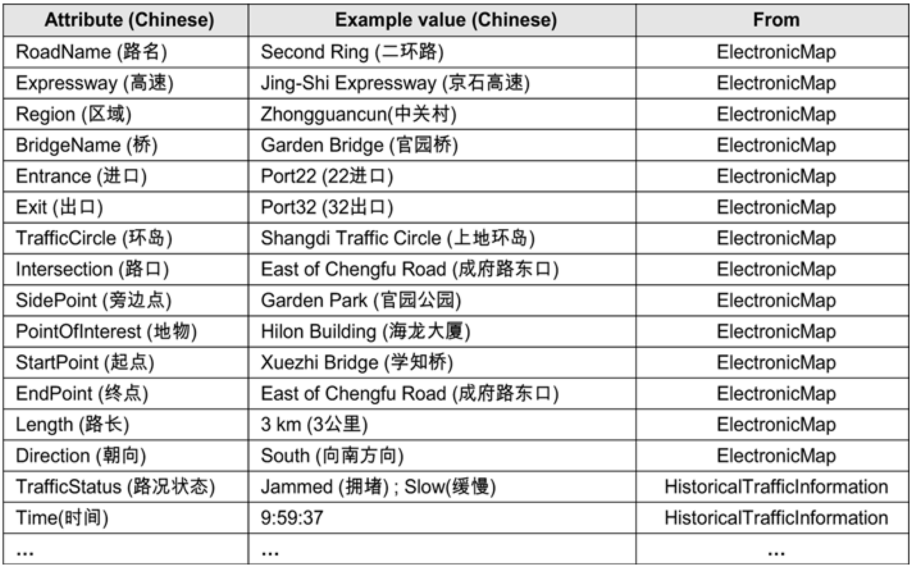
“:type” indicates the type of values an attribute takes, and ChineseString is the most commonly used value type.

“:from” is a device for indicating sources where the attribute-value pair is from. In the map ontology, the sources primarily include the electronic map and historical traffic information.

We developed each of the “Concepts (Attributes definition)” (see Fig. 8), “Relations” (see Fig. 9) and “Axioms” (see Fig. 10) aspects in detail.

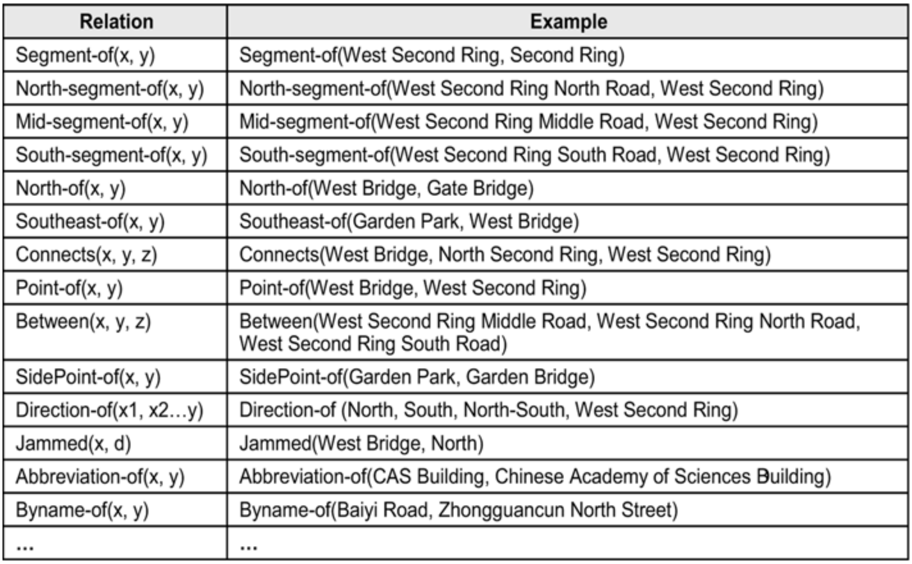
```
DefOntology MapOntology
{
       // Concepts
        Attribute: RoadName
                   :type ChineseString
                   :from ElectronicMap
        Attribute: BridgeName
                   :type ChineseString
                   :from ElectronicMap
        Attribute: TrafficStatus
                   :type ChineseString
                   :from HistoricalTrafficInformation
        ...
       // Relations
        Relation: Segment-of
                  :type (ChineseString, ChineseString)
        Relation: Point-of
                  :type (ChineseString, ChineseString)
        ...
       // Axioms
        Axiom: ∀x, y, z: Segment-of(x, y) & Segment-of(y, z) → Segment-of(x, z)
        Axiom: ∀x, y, z: Point-of(x, z) & Point-of(y, z) → Segment-of([x, y], z)
        Axiom: ∀x, y, z: Segment-of(x, y) & Point-of(z, x) → Point-of(z, y)
        ...
}
```

Fig. 7. Frame-Oriented Map Ontology



Attribute (Chinese)	Example value (Chinese)	From
RoadName (路名)	Second Ring (二环路)	ElectronicMap
Expressway (高速)	Jing-Shi Expressway (京石高速)	ElectronicMap
Region (区域)	Zhongguancun(中关村)	ElectronicMap
BridgeName (桥)	Garden Bridge (官园桥)	ElectronicMap
Entrance (进口)	Port22 (22进口)	ElectronicMap
Exit (出口)	Port32 (32出口)	ElectronicMap
TrafficCircle (环岛)	Shangdi Traffic Circle (上地环岛)	ElectronicMap
Intersection (路口)	East of Chengfu Road (成府路东口)	ElectronicMap
SidePoint (旁边点)	Garden Park (官园公园)	ElectronicMap
PointOfInterest (地物)	Hilon Building (海龙大厦)	ElectronicMap
StartPoint (起点)	Xuezhi Bridge (学知桥)	ElectronicMap
EndPoint (终点)	East of Chengfu Road (成府路东口)	ElectronicMap
Length (路长)	3 km (3公里)	ElectronicMap
Direction (朝向)	South (向南方向)	ElectronicMap
TrafficStatus (路况状态)	Jammed (拥堵) ; Slow(缓慢)	HistoricalTrafficInformation
Time(时间)	9:59:37	HistoricalTrafficInformation
...	...	...

Fig. 8. Attributes in Map Ontology

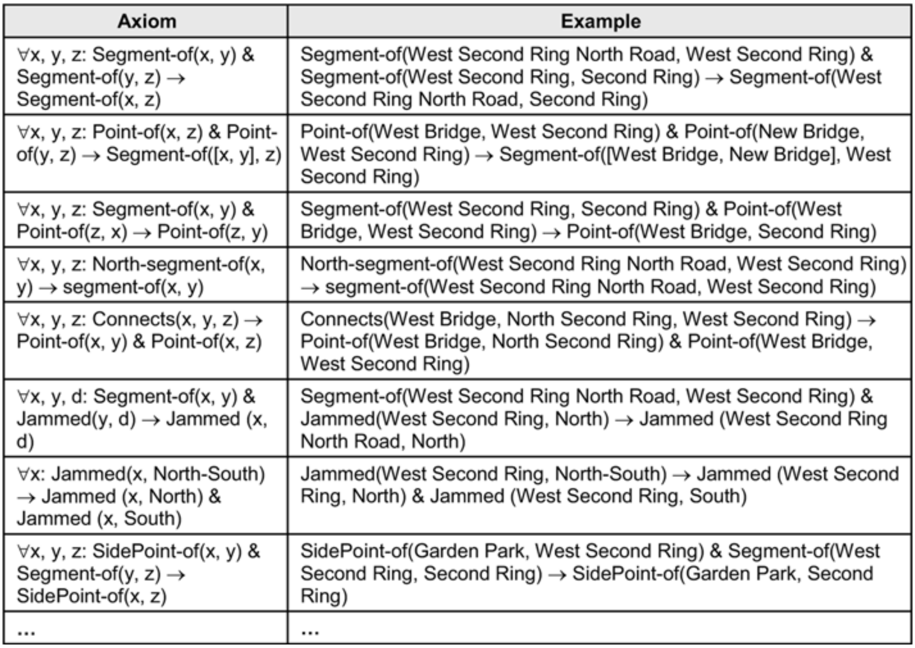


Relation	Example
Segment-of(x, y)	Segment-of(West Second Ring, Second Ring)
North-segment-of(x, y)	North-segment-of(West Second Ring North Road, West Second Ring)
Mid-segment-of(x, y)	Mid-segment-of(West Second Ring Middle Road, West Second Ring)
South-segment-of(x, y)	South-segment-of(West Second Ring South Road, West Second Ring)
North-of(x, y)	North-of(West Bridge, Gate Bridge)
Southeast-of(x, y)	Southeast-of(Garden Park, West Bridge)
Connects(x, y, z)	Connects(West Bridge, North Second Ring, West Second Ring)
Point-of(x, y)	Point-of(West Bridge, West Second Ring)
Between(x, y, z)	Between(West Second Ring Middle Road, West Second Ring North Road, West Second Ring South Road)
SidePoint-of(x, y)	SidePoint-of(Garden Park, Garden Bridge)
Direction-of(x1, x2…y)	Direction-of (North, South, North-South, West Second Ring)
Jammed(x, d)	Jammed(West Bridge, North)
Abbreviation-of(x, y)	Abbreviation-of(CAS Building, Chinese Academy of Sciences Building)
Byname-of(x, y)	Byname-of(Baiyi Road, Zhongguancun North Street)
...	...

Fig. 9. Relations in Map Ontology

3.4 Creating Map Ontology by GIS

The map ontology creator works based on functions of GIS[12,13,14], which can be used to extract the values of attributes in Fig. 8, and instance relations in Fig. 9.



Axiom	Example
∀x, y, z: Segment-of(x, y) & Segment-of(y, z) → Segment-of(x, z)	Segment-of(West Second Ring North Road, West Second Ring) & Segment-of(West Second Ring, Second Ring) → Segment-of(West Second Ring North Road, Second Ring)
∀x, y, z: Point-of(x, z) & Point-of(y, z) → Segment-of([x, y], z)	Point-of(West Bridge, West Second Ring) & Point-of(New Bridge, West Second Ring) → Segment-of([West Bridge, New Bridge], West Second Ring)
∀x, y, z: Segment-of(x, y) & Point-of(z, x) → Point-of(z, y)	Segment-of(West Second Ring, Second Ring) & Point-of(West Bridge, West Second Ring) → Point-of(West Bridge, Second Ring)
∀x, y, z: North-segment-of(x, y) → segment-of(x, y)	North-segment-of(West Second Ring North Road, West Second Ring) → segment-of(West Second Ring North Road, West Second Ring)
∀x, y, z: Connects(x, y, z) → Point-of(x, y) & Point-of(x, z)	Connects(West Bridge, North Second Ring, West Second Ring) → Point-of(West Bridge, North Second Ring) & Point-of(West Bridge, West Second Ring)
∀x, y, d: Segment-of(x, y) & Jammed(y, d) → Jammed (x, d)	Segment-of(West Second Ring North Road, West Second Ring) & Jammed(West Second Ring, North) → Jammed (West Second Ring North Road, North)
∀x: Jammed(x, North-South) → Jammed (x, North) & Jammed (x, South)	Jammed(West Second Ring, North-South) → Jammed (West Second Ring, North) & Jammed (West Second Ring, South)
∀x, y, z: SidePoint-of(x, y) & Segment-of(y, z) → SidePoint-of(x, z)	SidePoint-of(Garden Park, West Second Ring) & Segment-of(West Second Ring, Second Ring) → SidePoint-of(Garden Park, Second Ring)
...	...

Fig. 10. Axioms in Map Ontology

For example, the GIS function corresponding to the instance relation "Point-of (x, y)" in Fig. 9 in SuperMap[12] is "soDatasetVector.QueryEx(objGeometry As soGeometry, scsCommonPoint, "") as soRecordset", which is used to query the point (Bridge) on a line (Road) in the electronic map. Besides, both the values (traffic-status descriptions) of the attribute "TrafficStatus" in Fig. 8 and the axioms in Fig. 10 are primarily organized manually.

4 Map Ontology Driven Traffic Information Processing

4.1 Properties of Natural Language Traffic Information

Generally, most of reported direct, indirect and connotative traffic information can be organized in canonical structures, which falls into one of the following four formats:

(1) "where, traffic-status", such as "West Bridge is jammed", where "where" means the location of traffic congestion;

(2) "where, direction, traffic-status", such as "West Bridge, north bound, is jammed";

(3) "time, where, traffic-status", such as "9:59:37, West Bridge is jammed";

(4) "time, where, direction, traffic-status", such as "9:59:37, West Bridge, north bound, is jammed".

It should be noted that the “where”, “direction”, “traffic-status” and “time” permits sequence variations. For example, “The north bound direction of West Bridge is jammed”.

All of the above four formats are expressed as simple-sentence styled traffic information. In reality, complex-sentence styled traffic information is much more prevalent, and they are commonly expressed as combinations of the above four simple-sentence formats (see Fig. 14).

4.2 Traffic Information Processing in TIPL Based on Map Ontology

As shown above, most of the traffic information can be organized in quite canonical structures. This motivated the design of a traffic information processing language (TIPL) for use by engineers to write declarative programs for processing traffic info.

As shown in Fig. 11, TIPL, which works based on map ontology, comprises two parts:

(1) Syntax is a grammatical definition system, which records all kinds of syntaxes used to parse traffic info. It should be noted that Syntax is defined in a nested way.

(2) Agent is a collection of traffic information processing operations. Each agent has two major components. The first one is a context in which the agent can be activated. The second one is information processing operations. Each operation comes with a condition and an accompanying action. The condition of an operation is different from the agent’s context: the context represents a general situation for the agent (and operations) to be activated, whereas the condition of a concrete operation represents a more specific situation in the context where the operation can be activated to perform information processing.

In Fig. 11, for example, <syntaxi> is a context for op1 to opk, and <syntaxi> is further defined in Syntax part. When a syntax is used to define an agent, its parametric non-terminals can be instantiated, such as <?X>; thus we obtain a concrete agent with concrete behaviors. This kind of parameterization makes the syntax reusable not only across different information sources but also across different domains. Before executing an operation in the context, the agent checks whether its condition is met. If so, it performs the action; otherwise it executes the next operation, till all the operations were carried out.

Using TIPL, there are two steps to process traffic information:

(1) Parsing the syntax of traffic information, which means segmenting the sentence firstly and then dividing it into “Where”, and “Direction”, “TrafficStatus”, “Time” parts defined in Fig. 3.

(2) Analyzing the semantic meaning of the “Where” part, which means extracting the actual location(s) of congestion, that is, the value(s) of “RoadName” and “Jam-Point” defined in Fig. 3.

Example: Fig. 12 shows a portion of a TIPL program, which can be used to process information like “9:59:37 West Second Ring, North bound, is in slow traffic”, where:

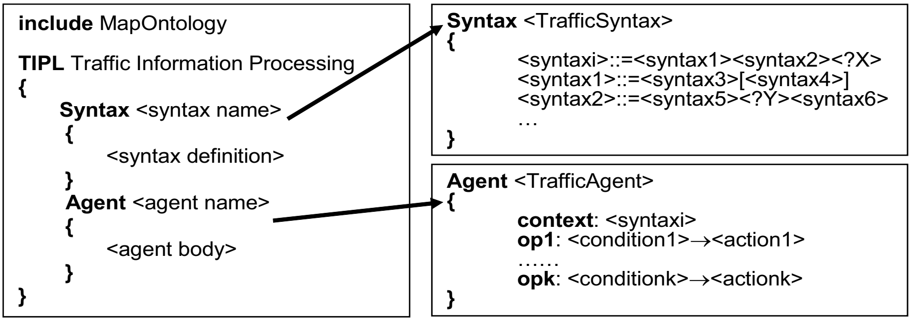

include MapOntology
TIPL Traffic Information Processing
{
Syntax <syntax name>
{
<syntax definition>
}
Agent <agent name>
{
<agent body>
}
}
Syntax <TrafficSyntax>
{
<syntaxi>::=<syntax1><syntax2><?X>
<syntax1>::=<syntax3>[<syntax4>]
<syntax2>::=<syntax5><?Y><syntax6>
…
}
Agent <TrafficAgent>
{
context: <syntaxi>
op1: <condition1>→<action1>
……
opk: <conditionk>→<actionk>
}

Fig. 11. Architecture of TIPL

- The brace pair means this syntax part is optional.
- "|" means "or" logical operator.
- forall(<str>) means that, for all instances of <str>, a certain action is performed.

First, it can be parsed from within the "Syntax" part that "9:59:37 West Second Ring, North bound, is in slow traffic" is suitable for "<SyntaxTypeA>", where "<Where>" is matched as "<ConnotativeLocation>". During this step, the <Attribute, Value> pairs—<RoadName, West Second Ring>, <Direction, north>, <TrafficStatus, slow> and <Time, 9:59:37>—in map ontology are referenced.

Second, it can be found out from within the "Agent" part that the context "<SyntaxTypeA>" is chosen, and "<ConnotativeLocation>" is met in op3. So the action "ConnotativeSemantic" is performed. It infers (decomposes) the connotative location "West Second Ring" as "West Bridge", "Garden Bridge", "Gate Bridge" and "New Bridge", and outputs the results as in Fig. 4. During this action, the relations Point-of(West Bridge, West Second Ring), Point-of(Garden Bridge, West Second Ring), etc., in map ontology are referenced.

5 Map Ontology Driven User Query Processing

As described in Section 2, there are three categories of traffic information queries: direct queries, indirect queries and connotative queries. Based on map ontology, these queries are also processed by a TIPL program. The program works in a way same to that of traffic information processing and will not introduce in detail in this paper.

6 Implementation and Evaluation

6.1 Implementation

The implementation of TISS consists of five components (Fig. 13):

```
include MapOntology

TIPL Traffic Information Processing
{
  Syntax <TrafficSyntax>
  {
    …
    <SyntaxTypeA>::= [<Time>]<Where><TrafficStatus>[<…>][<…>]
    <Where>::= <DirectLocation> | <IndirectLocation> | <ConnotativeLocation>
    <DirectLocation>::=<BridgeName>[<Direction>]
    <IndirectLocation>::=<SidePoint>[<Direction>]
    <ConnotativeLocation>::=<RoadName>[<Direction>]
    …
  }
  Agent <TrafficAgent>
  {
    context: <SyntaxTypeA>
    op1: forall (<DirectLocation>)→DirectSemantic(RoadName, JamPoint, Direction, TrafficStatus, Time)
    op2: forall (<IndirectLocation>)→IndirectSemantic(RoadName, JamPoint, Direction, TrafficStatus, Time)
    op3: forall (<ConnotativeLocation>)→ConnotativeSemantic(RoadName, JamPoint, Direction, TrafficStatus, Time)
    …
  }
}
```

Fig. 12. A portion of TIPL Program Processing Traffic Information

- Information Sources, which are divided into two categories. The first one is official data from Traffic Management Bureau, and the other one is traffic short messages reported by mobile users.
- Content Providers (CPs), who provide basic information for the service, such as electronic maps, traffic rule databases, historical traffic information, etc..
- Mobile Users, who report/query traffic information via SMS.
- Mobile Carriers (MCs) like China Mobile and China Unicom. MCs transmit queries and system response between Mobile Users and Service Providers.
- Service Providers (SPs), which are the key part of the traffic information service system. They work between CPs and MCs.

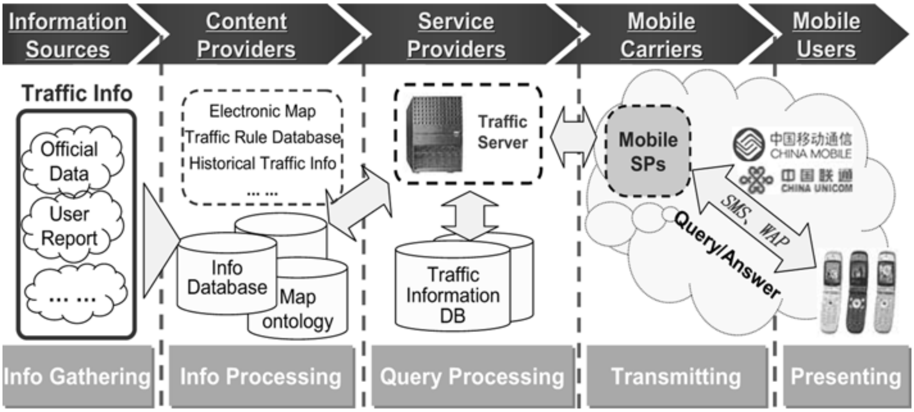
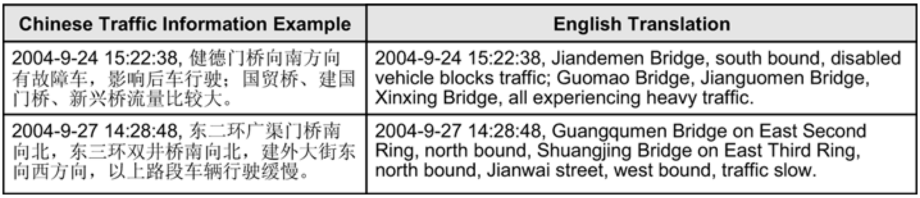


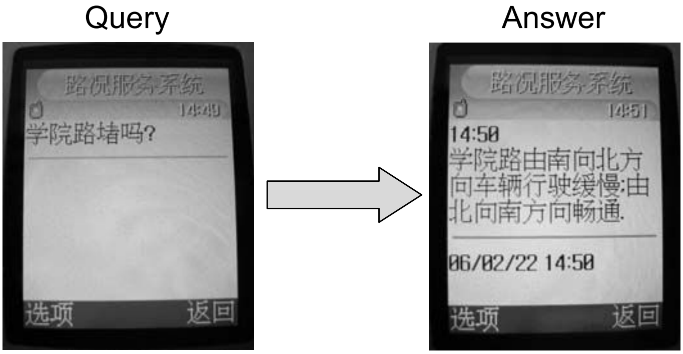
Fig. 13. Implementation of TISS



Chinese Traffic Information Example	English Translation
2004-9-24 15:22:38, 健德门桥向南方向有故障车，影响后车行驶；国贸桥、建国门桥、新兴桥流量比较大。	2004-9-24 15:22:38, Jiandemen Bridge, south bound, disabled vehicle blocks traffic; Guomao Bridge, Jianguomen Bridge, Xinxing Bridge, all experiencing heavy traffic.
2004-9-27 14:28:48, 东二环广渠门桥南向北，东三环双井桥南向北，建外大街东向西方向，以上路段车辆行驶缓慢。	2004-9-27 14:28:48, Guangqumen Bridge on East Second Ring, north bound, Shuangjing Bridge on East Third Ring, north bound, Jianwai street, west bound, traffic slow.

Fig. 14. Examples of Chinese Traffic Information Messages

Fig. 14 shows some examples of Chinese traffic information messages and their English translation. Fig. 15 shows an actual query process using TISS via SMS.



学院路堵吗?

14:50
学院路由南向北方向车辆行驶缓慢:由北向南方向畅通.

06/02/22 14:50

Fig. 15. Actual Traffic Information Query Process

6.2 Experiment 1: Accuracy Evaluation

Because the inputs to TISS are natural language traffic info reports and queries, the first experiment was used to investigate the accuracy of TIPL in processing them.

There are three categories of reports (traffic information): direct, indirect and connotative ones. We correspondingly defined three kinds of accuracies respectively as follows (corresponding to the inputs and analysis results in Fig. 3):

$$\begin{array}{l}\text{Accuracy for direct (indi-}\\ \text{rect,connotative) report}\end{array} = \frac{\begin{array}{c}\text{Number of reports whose terms \{RoadName, JamPoint,}\\ \text{Direction, TrafficStatus, Time\} are correctly extracted}\end{array}}{\text{Number of direct (indirect, connotative) reports}}$$

Likewise, we also defined the accuracy for direct (indirect or connotative) queries as follows (corresponding to the inputs and analysis results in Fig. 5):

$$\begin{array}{l}\text{Accuracy for direct (indi-}\\ \text{rect, connotative) query}\end{array} = \frac{\begin{array}{c}\text{Number of queries whose terms \{RoadName,}\\ \text{JamPoint, Direction\} are correctly extracted}\end{array}}{\text{Number of direct (indirect, connotative) queries}}$$

For the experimental dataset:

- We selected 20000 reports, including 8000 direct reports, 5000 indirect reports and 7000 connotative reports.
- We collected 2000 queries from mobile users, including 800 direct queries, 500 indirect queries and 700 connotative queries.

Accuracy Service	Direct	Indirect	Connotative	Average
Report	95%-97%	91%-93%	93%-95%	93%-95%
Query	97%-100%	93%-96%	95%-98%	95%-98%

Fig. 16. Accuracies for Reports and Queries Processing by TISS

Fig. 16 shows the results of the experiment, along with the average accuracies for the reports and queries processed. Note that the accuracies for indirect and connotative reports were basically 0% with other methods[5]. In contrast, the new method proposed in this paper achieved high accuracies for indirect and connotative reports as well as direct reports. However, there were 5%–7% errors for reports processed and 2%–5% errors for queries processed with our proposed method, and the reasons for these errors were:

(1) New traffic-status descriptions that were not collected into the map ontology were used in reports or queries.

(2) City extension generates new jam points. These new jam points that were not collected into the map ontology were reported or queried.

(3) Some wrongly written or mispronounced characters existed in the reports or queries.

6.3 Experiment 2: Performance Evaluation

The second experiment was used to examine the performance of TISS.

Performance Service	Scalability	Response time
Report	7X15 (Thread) Requests/Second	150 Millisecond/Request
Query	10X15 (Thread) Requests/Second	100 Millisecond/Request

Fig. 17. Performance Evaluation on TISS

We deployed TISS on a HP DL580 server with 2200MHz CPU×4 and 4GB memory. Then we used 6 million pieces of historical traffic information and 2000 queries repeatedly 10 times (20000 queries in all) to test the performance of the system, and the scalability and response time evaluations are shown in Fig. 17, where scalability represents the ability of TISS to process how many requests in a certain time interval, and response time measures the delay between a request and its answer.

7 Conclusion

In this paper, we supply a new approach for gathering traffic information by mobile users reporting via short message service. We also supply a traffic information service system providing mobile users the opportunity to query traffic information in natural language. Then a method is designed to deal with the direct, indirect, connotative traffic information and queries based on map ontology. Our proposed method can improve the natural language processing accuracy to 93%–95% specific to the traffic information domain.

References

1. Amir Reza Mamdoohi, Mohammad Kermanshah.: Traffic Information Use Modeling in the Context of a Developing Country. PERIODICA POLYTECHNICA SER. TRANSP. ENG. VOL. 33, NO. 1-2 (2005) 125-137
2. R. Chrobok, S.F. Hafstein, A. Pottmeier.: OLSIM: A New Generation of Traffic Information Systems. In: Forschung und wissenschaftliches Rechnen, GWDG-Bericht Nr. 63, Eds. V. Macho and K. Kremer (2004) 11-25
3. Takayuki Nakata, Jun-ichi Takeuchi.: Mining Traffic Data from Probe-Car System for Travel Time Prediction. Proc. of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA (2004) 817-822
4. G. Peersman, et al.: A tutorial overview of the short message service within GSM. Computing and Control Engineering Journal (2000) 79-89
5. R. Evans, R. Gaizauskas, L.J. Cahill, J. Walker, J. Richardson, A. Dixon.: POETIC: A System for Gathering and Disseminating Traffic Information. Journal of Natural Language Engineering 1(4) (1995) 363-387
6. T. R. Gruber.: A Translation Approach to Portable Ontology Specification. Knowledge Acquisition, 5(2) (1993) 199-220
7. http://www.w3.org/TR/owl-features/
8. Cungen Cao, Haitao Wang, Yuefei Sui.: Knowledge modeling and acquisition of traditional Chinese herbal drugs and formulae from text. Artificial Intelligence in Medicine (2004) 32, 3-13
9. Cungen Cao, Qiangze Feng, et al.: Progress in the Development of National Knowledge Infrastructure. Journal of Computer Science and Technology, vol.17 (2002) 523-534
10. Lars Kulik, Matt Duckham, Max Egenhofer.: Ontology-Driven Map Generalization. Journal of Visual Languages and Computing 16 (3) (2005) 245-267
11. Manfred M. Fischer, Peter Nijkamp.: Geographic Information Systems, Spatial Modeling and Policy Evaluation. Berlin: Springer-Verlag (1993)
12. Understanding SuperMap GIS. SuperMap GIS Technologies, Inc. Beijing (2003). http://www.supermap.com.cn/downloadcenter/download.asp?cur_page=18
13. Ian Johnson. Understanding MapInfo: A Structured Guide (1996) 300pp, 225 ill. ISBN 1864510161. Published by the Archaeological Computing Laboratory, University of Sydney. http://www.mapinfo.com/
14. ESRI, Understanding GIS–The ArcInfo Method. Cambridge, United Kingdom, UK: GeoInformation International (1997). http://www.arcinfo.com/
15. Martien Molenaar.: An Introdcution to the Theory of Spatial Object Modelling, London:Taylor & Francis Ltd (1998)

A Knowledge- and Workflow-Based System for Supporting Order Fulfillment Process in the Build-to-Order Supply Chains

Yan Ye[1, 2], Dong Yang[1], Zhibin Jiang[1], and Lixin Tong[1]

[1] Department of Industrial Engineering and Management, Shanghai Jiao Tong University,
1954 Hua Shan Road, 200030 Shanghai, China
yeyna@sjtu.edu.cn, dongyangcn@hotmail.com
zbjiang@sjtu.edu.cn, culizn@163.com
[2] College of Mechatronics Engineering, Zhejiang University of Technology,
310032 Hangzhou, China
yeyan_yan@yahoo.com.cn

Abstract. A web-based system for order fulfillment provides a software environment for successfully implementing the build-to-order supply chain (BOSC) strategy. However, current efforts in this domain are not adequate to support automatic reasoning of knowledge-intensive activities within order processing processes and supervision and flexible responsiveness of the entire workflows. Based on the application scenario from a conveyer BOSC, this paper proposes an approach to developing an order management system called OMS-KW enhanced by the Semantic Web and workflow technologies. A multi-ontology-based approach is presented to facilitate representation, sharing and reuse of different types of knowledge. Moreover, multiple ontologies in OWL provide semantic foundation for interoperability between the system and other systems. In addition, problem-solving methods (PSMs) and SWRL-based rules are developed to enable automatic execution of knowledge-intensive activities. Furthermore, the system integrates the knowledge and workflow applications to monitor all the knowledge-intensive and non-knowledge-intensive activities and to improve system flexibility.

Keywords: Ontologies, PSMs, Workflow, Order Fulfillment Process (OFP), BOSC.

1 Introduction

In the market environments characterized by increasingly diverse customer requirements and fierce global competitions, customer-centric enterprises are actively organizing the build-to-order supply chain (BOSC) [1] to secure market shares and improve organizational competitiveness. A BOSC is a value chain that leverages information technology and strategic alliances based on core competencies to build the products satisfying individual customer requirements [2]. Its main objective is to achieve the greatest degree of responsiveness to changing customer needs in a cost-effective manner. Customer requirements are normally embodied by orders that are

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 711 – 724, 2006.
© Springer-Verlag Berlin Heidelberg 2006

the beginning and the end of enterprise operations and BOSC management. Therefore, the objective of the BOSC at the tactical level can be described as efficiently fulfilling the orders of individual customers.

An order fulfillment process (OFP) starts with receiving orders from customers and ends with having the finished goods delivered [3]. The process is composed of various interdependent activities. Some of them are performed by workers using knowledge, such as design knowledge and enterprise capabilities. These activities are called knowledge-intensive activities (KIAs) [4]. For example, typical knowledge-intensive activities include evaluation, classification and configuration. The performance of these activities has a great influence on the effectiveness of the OFP. In the BOSC, ordered products are tailored specifically to customer needs and are of high variety. These make knowledge and their internal relationships involved in the knowledge-intensive activities complicated and in turn make these activities difficult to execute manually and efficiently. Thus, these activities often become the performance bottlenecks of the OFP and supporting their automatic execution with the help of knowledge-based order management systems is necessary. In addition, the OFP in the BOSC is cooperatively realized by organizationally independent and geographically distributed enterprises within strategic alliances. To efficiently meet changing customer needs, these enterprises are obliged to manage, monitor and integrate the entire OFP. On the one hand, this requires enterprises to timely exchange and share information related to orders with a common understanding of the semantics associated with the information. However, effective information exchange may be hindered by semantic discrepancies and clashes among enterprises. On the other hand, it is necessary to rapidly adjust and optimize business processes according to the changes reflected by the information. However, the lack of flexible response to changes may impair customer service quality and enterprise profitability. For example, Apple Computer was unable to fill orders for its new high-end line of G4 computers and experienced a devastating 14 percent drop in revenue in 1999, because it was not aware of delays in chip supplies and did not make adjustments to compensate for the delays as they occurred [5]. Therefore, a knowledge-based system supporting order processing processes in the BOSC need to provide formal semantics of knowledge explicitly for its interaction with legacy systems in enterprises through Internet, and at the same time, to provide capabilities to supervise the entire processes and quickly response to changes. However, current efforts in order management systems have not been found to meet the requirements mentioned above.

In this paper, we propose an approach to developing a knowledge-based order management system called OMS-KW enhanced by the Semantic Web and workflow technologies [6, 7] that is applied to a conveyer BOSC scenario. Firstly, a multi-ontology-based method is presented to provide a common semantic framework for the system and enable semantic interoperability between the system and other systems. Secondly, problem-solving methods (PSMs) modules and relevant rules are developed to automatically perform knowledge-intensive activities in the OFPs and to provide reusability and maintainability for the OMS-KW. Thirdly, a knowledge application is developed to initiate and supervise the reasoning of knowledge-intensive activities and is combined with the workflow management system (WFMS) that models, manages and monitors the entire OFPs.

The remainder of the paper is organized as follows. An overview of related work is given in Section 2. Section 3 presents the application scenario and architecture of the OMS-KW. In Section 4, a multi-ontology-based approach is explained. Section 5 describes the process modeling and analysis method for the OMS-KW. The PSMs modules for knowledge-intensive activities are discussed in Section 6. Finally, conclusions are given.

2 Related Work

There have been several efforts in the system development for OFPs. Badell and Puigjaner [8] have developed a web-based autonomous order entry system for companies in process industries. Balve et al. [9] have described an order management system for transformable manufacturing enterprises based on system theory and management cybernetics. These systems are not specially focused on the BOSC domain. Moreover, they less consider the reasoning support of knowledge-intensive activities and do not really leverage the potential of the Semantic Web technologies.

The integration of workflow and knowledge management has been presented in some other systems, such as VirtualOffice, KnowMore and KnowledgeScope. VirtualOffice [10] uses a document analysis and understanding (DAU) system as a workflow application to transfer information in paper documents into the electronic information that is integrated into the workflows. The KnowMore [10] project develops information agents for the ontology-based, heuristic information retrieval to proactively provide context-sensitive information to the workers who execute relevant tasks within a workflow. KnowledgeScope [11] is a knowledge management system that captures and retrieves knowledge as a workflow proceeds and that organizes the knowledge and context in a knowledge repository based on a proposed process meta-model. The main objective of these systems is to support people who perform knowledge-intensive activities in the workflow by providing knowledge, while the OMS-KW aims to support the integration of the automatic reasoning of knowledge-intensive activities with the WFMS.

The approach to developing the OMS-KW has many similarities with the CommandKADS knowledge engineering methodology [4] and the UPML architecture [12] in the areas of knowledge-based systems. The CommandKADS project develops the *model of expertise* consisting of domain, inference, and task layers, each of which describes specific aspect of a knowledge-based system. UPML decomposes a knowledge-based system into related elements: tasks, problem-solving methods, domain models, ontologies and adapters. The contributions of the proposed approach are to deal with specific knowledge problems contained in the order fulfillment workflow of the BOSC domain using the Semantic Web technologies, especially ontologies and rules.

3 Application Scenario and System Architecture

3.1 Application Scenario

The application scenario used to exemplify the proposed approach is provided by a large company with a high market share for belt conveyers. The company has about 60 major suppliers and builds all conveyers based on customer specifications. Its basic business processes are described as following. After the sales staffs receive purchase requests from customers, the technical department personnel, interacting with customers, specify functional requirements and the conveyer's attributes that can be personalized, such as working environments and carrying materials. The technicians also obtain specific constraints and preferences. For example, the customer may demand the imported electromotor used in the conveyer. According to these requirements, the technicians design the technical details of the ordered product and send the assembly drawing to the customer. At the same time, the sales staffs consult order details and sign a formal contract with the customer. Finally, suppliers are selected to provide required outsourcing components. All components are assembled to the customized conveyer that is then tested at the working site appointed by the customer.

In the current OFP, the technicians spent much time on the confirmation of individual customer requirements that is in fact one of sales support activities. This detracts engineering resources from the development of new product families. Moreover, the technicians can not always assure effective and valid order configurations during the conveyers design because of complicated knowledge and constraint relationships existing in all kinds of conveyers as well as a large number of orders but limited time. In addition, the relationship between components and suppliers is often many-to-many, that is, a supplier provides many component types and one component type is offered by several suppliers. Furthermore, different suppliers have different capabilities to produce the same component type. Exact semantics of these knowledge and personalized order requirements may not be understood by the company personnel, which often decreases the efficiency of the supplier selection. Therefore, the company has planned to implement the BOSC strategy with the aim of responding to the requirements of individual customers efficiently and maximizing the supply chain benefits through strategic alliances with suppliers. A flexible and effective order management system is an important supporting technology for achieving the goal.

3.2 System Architecture

The proposed architecture of the OMS-KW is shown in Fig. 1. The system uses J2EE as the development platform and adopts the four-tier browser/server structure.

The top level is the client tier, a web-based user interface for different types of users. Through the interface, customers can query and retrieve product information, such as the types and specifications of conveyers, and customize and place conveyer orders. Collaborative enterprises can upload and modify knowledge about their capabilities, and exchange the order-related information with the OMS-KW.

Furthermore, based on the interface, system administrators and designers can define the process models of OFPs that are saved in the workflow model base and explained and executed by the workflow engine. In addition, knowledge engineers can manage, update and maintain all the ontologies, rules and knowledge bases.

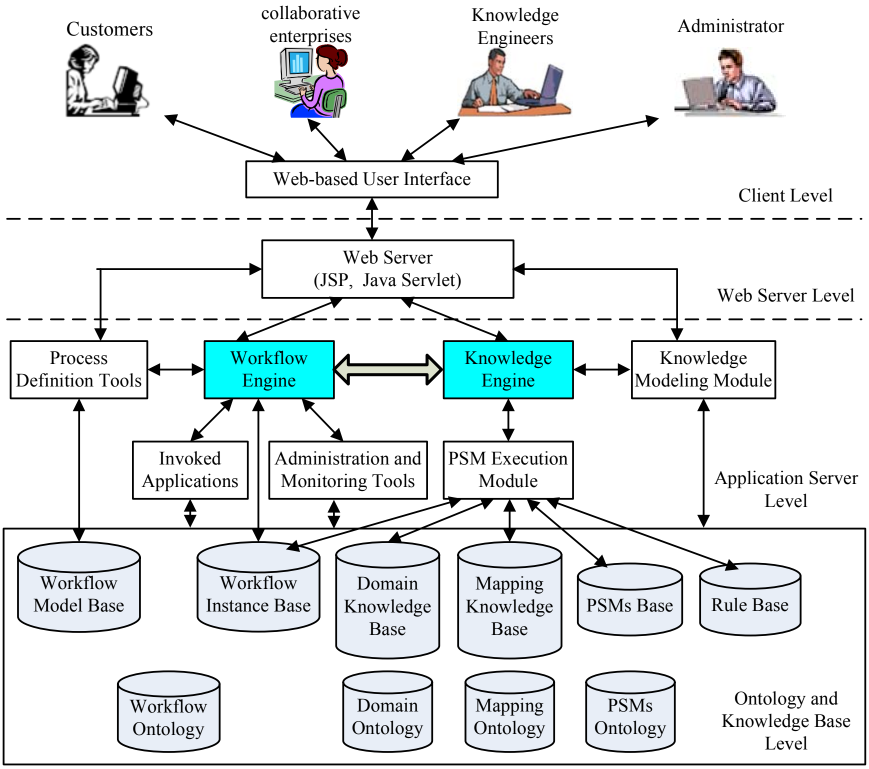


Fig. 1. The architecture of the OMS-KW

The second level is the web server that contains web components, such as JSP and Servlet. It receives requests from the client tier, invokes related system modules and returns the corresponding results to the web browser. For example, when a customer orders a customized conveyer, the web server initiates the workflow engine to instantiate a specific OFP.

The third level is the application server including the knowledge and workflow applications that mainly deals with core business logics of the OMS-KW. Customer requests for customized conveyers activate the workflow application. The workflow engine creates a process instance of processing the corresponding order and dynamically assigns activity instances in the process instance to the invoked applications within the BOSC enterprises. If a certain business activity is a knowledge-intensive activity, the workflow engine activates the knowledge engine. According to the information coming from the workflow engine, the knowledge engine invokes the PSM Execution Module to solve the problem. It also controls, manages and monitors the execution of knowledge-intensive activities and returns the reasoning results to the workflow engine.

The bottom level is the ontology and knowledge bases. The workflow model and instance bases save the OFPs knowledge, including all the knowledge-intensive and non-knowledge-intensive activities in the OFPs and logic dependency relationships among these activities. Multiple ontologies define basic concepts and relationships and constraints among these concepts in the order fulfillment applications of the BOSC. They provide terms for describing specific knowledge models and lay a good semantic foundation for the applications. The OWL language [13] is used to formally define these ontologies and knowledge. In addition, the rule base contains all the inference and business rules represented by the SWRL language [14].

The OMS-KW provides a single access interface for customers and all enterprises in the conveyer supply chain. It also links the company with its customers and partnering firms. On the one hand, through the OMS-KW, customers can easily access product information and order customized conveyers, and suppliers and logistics providers can offer the latest information on components and logistics services they provide at any moment. On the other hand, the company quickly configures suitable conveyers and builds supply chains to fulfill individual customer requirements by using the system. In addition, one of important characteristics of the OMS-KW is integrating knowledge reasoning capability with the WFMS. Thus, complex knowledge-intensive activities in the OFPs are performed automatically by the knowledge applications and the whole OFPs are managed and supervised by the WFMS.

4 A Multi-ontology-Based Approach

An ontology is a formal, explicit specification of a shared conceptualization [15]. It explicitly represents the concepts and relationships within a domain of discourse in a structural way and provides shared, formal semantic descriptions. Ontologies therefore can serve as the foundation for communication and sharing of knowledge among people and heterogeneous applications and for semantic interoperability in the context of the web. Moreover, ontologies provide semantic ground for the performance of reasoning tasks in applications. Therefore, a multi-ontology-based approach is proposed to provide a semantic framework for workflow operations and knowledge reasoning within the OMS-KW and for its interaction with other systems. The multi-ontology-based approach framework is shown in Fig. 2.

In Fig. 2, there are four types of ontologies that play different roles in building the OMS-KW models. The workflow ontology describes structural and control knowledge of workflow processes and enables semantic interoperability between workflow applications and other systems. The domain ontology provides the shared terms for modeling static knowledge of application domains. The PSMs ontology supports the descriptions of the competences and inference structures of PSMs that are used to perform knowledge-intensive activities, independent of any domain. In addition, inference rules used by PSMs are stored in the rule base. The separation of the descriptions of workflow processes, domains and PSMs facilitates knowledge sharing and reuse, that is, different PSMs can reason different knowledge-intensive activities in workflow processes by reusing different domain knowledge. However, it is necessary for an integrated system to build explicit inter-linkage or corresponding relationships among them. To this end, the mapping ontology provides structural ground for bridging conceptual and syntactic gaps among the workflow ontology, domain ontology, PSMs ontology and inference rules.

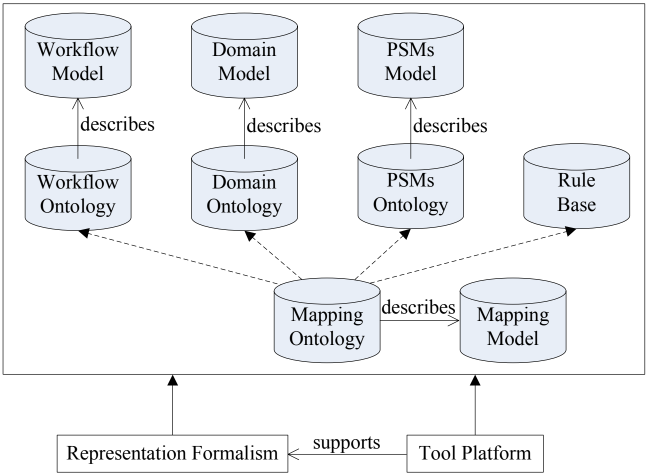


Fig. 2. The multi-ontology-based methodology framework

All the ontologies and knowledge are supposed to be described by the common knowledge representation formalism, such as OWL used in the OMS-KW. OWL [13] is proposed by the W3C organization to be the standardized and broadly accepted ontology language of the Semantic Web. The language is based on RDF/XML syntax and has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full. These languages provide formal semantics and automated reasoning support (e.g. class consistency and subsumption checking) through the mappings of OWL on predicate logics and description logics. In addition, tools supporting the formalism provide an environment for the descriptions of ontologies and knowledge. Protégé [16] is such a platform for constructing domain models and knowledge-based applications with ontologies. It provides a suite of tools, such as Protégé-OWL [17] and OWLViz [18] plugins, to support the creation, visualization, and manipulation of ontologies in various representation formats including RDF(S) and OWL. The Protégé-OWL plugin [17] enables users to build ontologies for the Semantic Web in the OWL. OWLViz [18] is designed to be used with the Protégé-OWL plugin and to view, incrementally navigate and compare the asserted class hierarchy and the inferred class hierarchy in an OWL Ontology. Fig. 3 shows a section of the class hierarchy of the domain ontology developed by using the Protégé-OWL and OWLViz plugins in the Protégé platform.

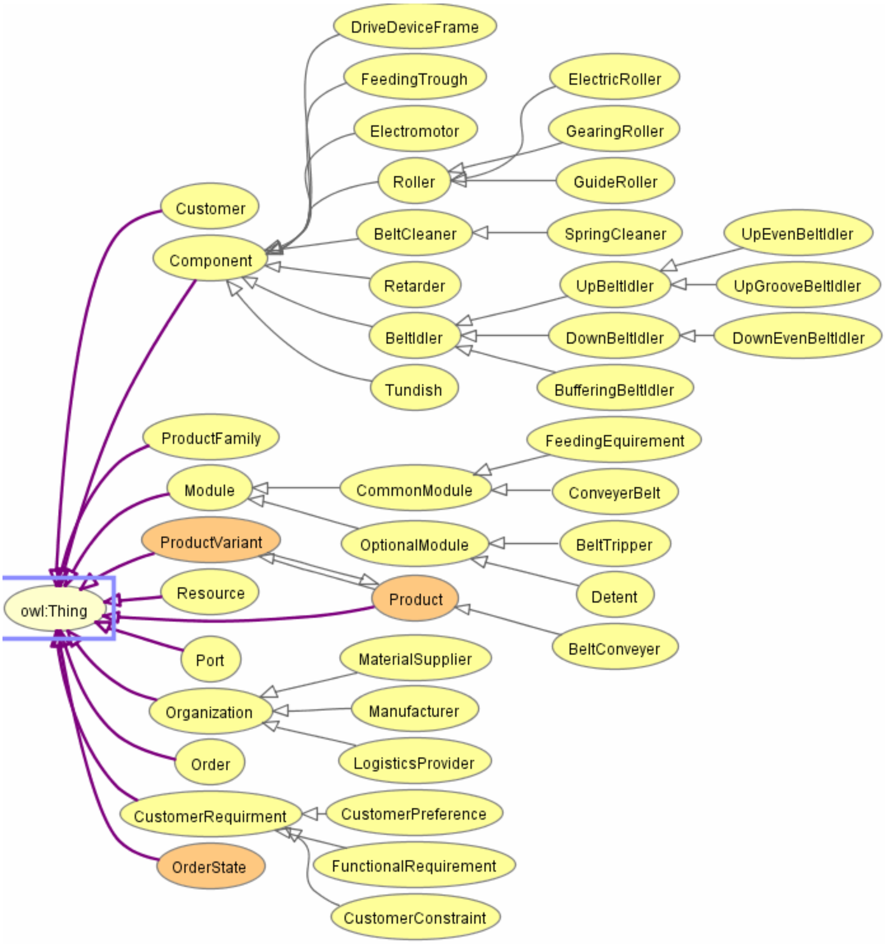


Fig. 3. Section of the class hierarchy of the OMS-KW domain ontology visualized using the OWLViz plugin

Basic classes used to describe product configuration knowledge include *ProductFamily*, *ProductVariant*, *Product*, *Module*, *Component*, *Port* and *Resource*. A product family contains some product variants. The class *ProductVariant* is equivalent with the *Product* class as they represent the product provider perspective and the customer perspective, respectively. A module is defined as a sub-system within a product family and may be decomposed to sub-modules and/or components. The class has two subclasses. One is the *CommonModule* class that is shared by all the product variants of a product family, such as conveyer belt and feeding equipment in the belt conveyer family. The other is the *OptionalModule* class that is used to satisfy specific customer requirements, such as detent. Modules and components can be connected by ports and can produce or consume resources. In addition, there exist *requires* and *incompatible_with* relations between components. The *requires* relation expresses that the usage of one component type needs the existence of the other component type, such as the relationship between gearing rollers and motors. The *incompatible_with* relation represents that two component types can not exist in a product variant at the same time, such as the relationship between electric rollers and gearing rollers. Part of OWL definitions of these classes and relations are shown in Fig. 4.

```
…
<owl:Ontology rdf:about=""/>
…
<owl:Class rdf:ID="ProductFamily">
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:onProperty >
            <owl:ObjectProperty rdf:ID="contains"/>
         </owl:onProperty>
         <owl:someValuesFrom rdf:resource="# ProductVariant"/>
      </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>
…
<owl:Class rdf:ID="Product">
   <owl:equivalentClass rdf:resource="#ProductVariant"/>
</owl:Class>
…
<owl:Class rdf:ID="BeltCleaner">
   <rdfs:subClassOf rdf:resource="#Component"/>
</owl:Class>
…
<owl:ObjectProperty rdf:ID="requires">
   <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="is_required_by"/>
   </owl:inverseOf>
   <rdfs:domain rdf:resource="#Component"/>
   <rdfs:range rdf:resource="#Component"/>
</owl:ObjectProperty>
```

Fig. 4. The OWL definitions of the OMS-KW domain ontology

The workflow ontology is developed based on the OWL-S [19]. OWL-S is an OWL-based web service ontology. Following the layered development approach, it contains several upper ontologies for services, one of which is the process ontology. The ontology defines terms for describing processes and their control structures. Considering the requirements of the OMS-KW, the workflow ontology imports the process ontology and makes some extensions. For example, two disjoint classes. *KnowledgeIntensiveActivity* and *NonKnowledgeIntensiveActivity* are defined as the subclasses of the class *Process* in the process ontology. The *KnowledgeIntensiveActivity* class is a set of all knowledge-intensive activities. Its definition is as follows.

```
<owl:Class rdf:ID="KnowledgeIntensiveActivity">
  <rdfs:subClassOf rdf:resource="&process;#Process"/>
  <owl:disjointWith>
    <owl:Class rdf:ID="NonKnowledgeIntensiveActivity "/>
  </owl:disjointWith>
</owl:Class>
```

5 Modeling and Analysis of the Conveyer Order Processing Workflow

The OMS-KW integrates workflow and knowledge-based reasoning to carry out flexible, efficient configuration and processing of the orders of customized products and to rapidly build supply chain systems. Workflow is the computerized facilitation or automation of a business process, in whole or part [7]. Its explicit modeling and analysis are the basis of the implementation of the system. Knowledge-based reasoning is to deal with complex problems, such as knowledge-intensive activities, by making use of knowledge [12]. The key requirement for the integration of workflow and knowledge reasoning is to model the order processing process and analyze and identify knowledge-intensive activities. UML [20] provides an intuitive analysis tool for the requirement. Fig. 5 shows the conveyer order processing workflow model represented by the UML activity diagram with object flows.

The model describes a typical process of acquiring and configuring the orders of customized conveyers. After customers select conveyer customization and ordering through the interface provided by the OMS-KW, an interaction process of order acquisition between the system and customers begins. The process is represented by three activities: specify functional requirements, specify constraints and preferences, and place customized order, as shown in Fig. 5. On receiving orders, the OMS-KW configures the customized conveyer to produce a design solution, estimates delivery date and price on the basis of the solution and other knowledge on customers, modules and components, etc. to supplement order details, and then provides the solution and these details to the customer online for confirmation. During this process, the order state changes from “placed” to “designed” and “confirmed”. After the customer confirms his order, the OMS-KW determines supply sources of all the modules and components in the conveyer solution. For components provided by the company itself, such as idlers and rollers, the system generates production documentations and transfers them to existing systems in the company, for example, the inventory system. For outsourcing components, such as conveyer belts, the system retrieves knowledge on supplier competences from knowledge bases and evaluates and selects suppliers for each component type in order to form supply chain networks for the order fulfillment.

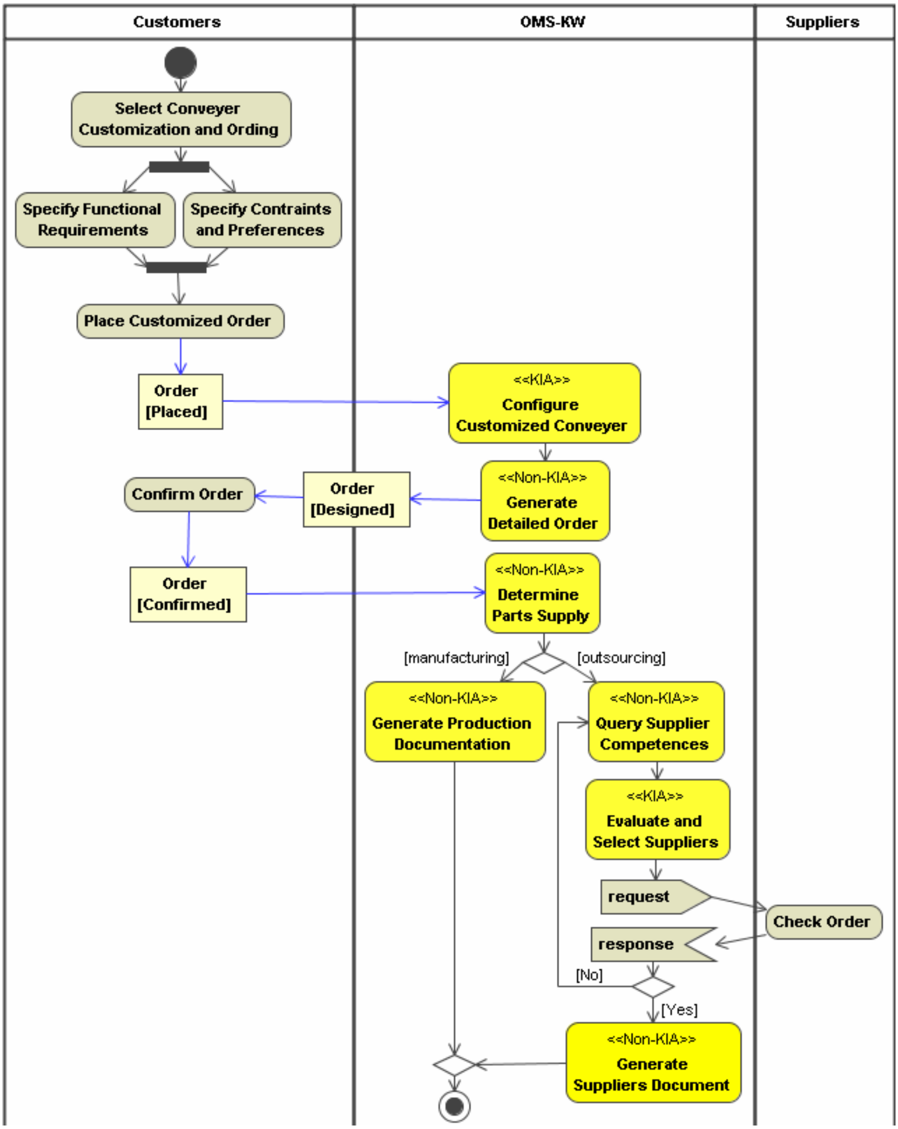

Customers
OMS-KW
Suppliers
Select Conveyer Customization and Ording
Specify Functional Requirements
Specify Contraints and Preferences
Place Customized Order
Order [Placed]
<<KIA>> Configure Customized Conveyer
<<Non-KIA>> Generate Detailed Order
Order [Designed]
Confirm Order
Order [Confirmed]
<<Non-KIA>> Determine Parts Supply
[manufacturing]
[outsourcing]
<<Non-KIA>> Generate Production Documentation
<<Non-KIA>> Query Supplier Competences
<<KIA>> Evaluate and Select Suppliers
request
Check Order
response
[No]
[Yes]
<<Non-KIA>> Generate Suppliers Document

Fig. 5. The conveyer order processing workflow model

By analyzing inputs, outputs, conditions, knowledge and resources needed for the execution of each activity in the process shown in Fig. 5, two activities are identified as knowledge-intensive activities and are defined as the subclasses of the *KnowledgeIntensiveActivity* class in the workflow ontology. One is the activity of configuring customized conveyer. During the conveyer configuration in terms of customer requirements, technicians normally determine required components, modules and their parameters by using a great deal of technical knowledge. The other activity is evaluating and selecting suppliers. The company personnel generally perform the activity based on the criteria of specific customer order and relevant knowledge such as brands and specifications of parts and supplier competences. In the OMS-KW, these activities are automatically performed by different PSM modules based on specifications in the mapping knowledge base.

6 PSMs for Knowledge-Intensive Activities

PSMs describe the strategies of solving knowledge-intensive problems that include types of knowledge, inference steps, and control flow between the inferences. Modeling PSMs knowledge in a domain-independent way enables the reuse of PSMs across different domains and knowledge-intensive activities. Thus, knowledge-based applications can use PSMs as reusable software components to maintain high flexibility and maintainability.

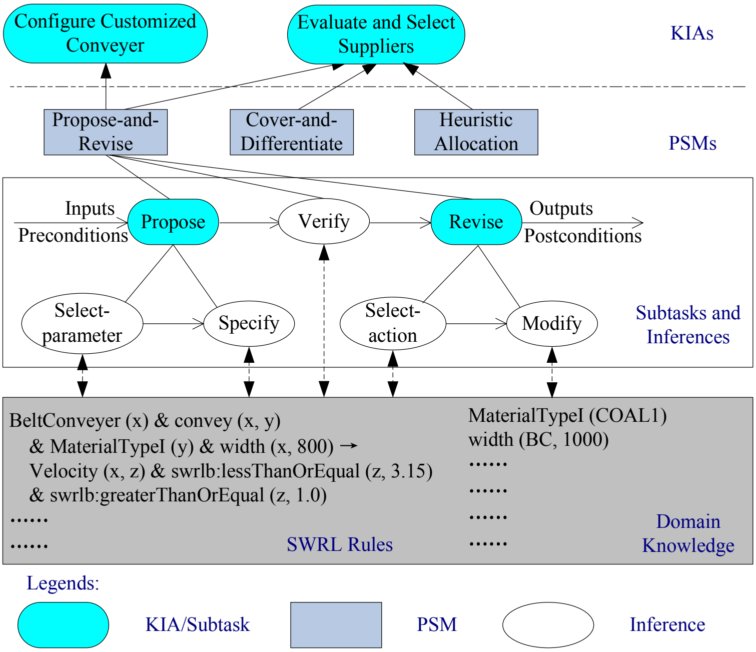


Fig. 6. PSMs reuse for knowledge-intensive activities in the OMS-KW

Two knowledge-intensive activities identified in Section 5 can be performed by different PSMs, as shown in Fig. 6. Reasoning objectives of both activities may be achieved by the *Propose-and-Revise* method. Moreover, according to different customer orders or business rules, the activity of evaluating and selecting suppliers may also be accomplished by the *Cover-and-Differentiate* or *Heuristic Allocation* method. Each PSM defines its inputs, outputs, preconditions, post-conditions and control structures with domain-independent terms. The control structure describes that which subtasks and inferences each method may be decomposed into and what order these subtasks and inferences follow. Inferences can be carried out directly by using available rule and fact knowledge. For instance, the *Propose-and-Revise* method in Fig. 6 is decomposed into five inferences: Select-parameter, Specify, Verify, Select-action and Modify, which are mapped to SWRL rules and domain knowledge. These SWRL rules and domain knowledge can be transformed into rules and facts in JESS language [21] to realize the inferences. Therefore, based on these active inferences and their control flow, the PSM can automatically execute the corresponding knowledge-intensive activities.

7 Conclusions

Efficient fulfillment of individual customer orders has become an important objective of the BOSC. However, it is difficult for current order management systems to support the reasoning of complicated knowledge-intensive activities and to provide capabilities of integrated management and quick responsiveness for the entire OFPs. As a significant effort, this paper presents a knowledge- and workflow-based order management system called OMS-KW enhanced by the Semantic Web technologies with the aim of meeting individual customer requirements efficiently and minimizing costs along value chains. Moreover, the application scenario from a conveyer BOSC is also described to show the effectiveness of the system. The semantic foundation for the interaction of the OMS-KW with other systems is the proposed OWL-based multi-ontology approach. The approach provides the workflow ontology, domain ontology and PSMs ontology to describe different types of knowledge in the OMS-KW and to facilitate knowledge sharing and reuse. In addition, the mapping ontology and corresponding knowledge base reconcile workflow, domain, PSMs and rule knowledge to form a knowledge-based application and to enable efficient, automatic execution of knowledge-intensive activities. Moreover, the OMS-KW combines the knowledge reasoning application with the workflow application to achieve integrated scheduling and supervision of knowledge-intensive activities and non-knowledge-intensive activities contained in the order fulfillment processes in the BOSC. Future works will include the enrichment and refinement of ontology-based rule and knowledge bases and the further development of PSM modules.

Acknowledgments. The work described in this paper was supported by a Research Grant from National Natural Science Foundation of China (Project No. 70471023).

References

1. Gunasekaran, A.: The Build-to-Order Supply Chain (BOSC): A Competitive Strategy for 21st Century. Journal of Operations Management 23 (2005) 419–422
2. Gunasekarana, A., Ngai, E.W.T.: Build-to-Order Supply Chain Management: A Literature Review and Framework for Development. Journal of Operations Management 23 (2005) 423–451

3. Lin, F.-R., Shaw, M.J.: Reengineering the Order Fulfillment Process in Supply Chain Networks. International Journal of Flexible Manufacturing Systems 10(3) (1998) 197 – 229
4. Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., van de Velde, W., Wielinga, B.: Knowledge Engineering and Management: The CommonKADS Methodology. The MIT Press, Cambridge Massachusetts London England (1999)
5. Ghiassi, M., Spera, C.: Defining the Internet-Based Supply Chain System for Mass Customized Markets. Computers and Industrial Engineering 45 (2003) 17–41
6. Antoniou, G., van Harmelen, F.: A Semantic Web Primer. The MIT Press, Cambridge Massachusetts London England (2004)
7. Hollingsworth, D.: The Workflow Reference Model. WFMC TC00-1003. 19 January 1995. http://www.wfmc.org/standards/docs/tc003v11.pdf
8. Badell, M., Puigjaner, L.: Advanced Enterprise Resource Management Systems for the Batch Industry. The TicTacToe algorithm. Computers & Chemical Engineering 25(4-6) (2001) 517-538
9. Balve, P., Wiendahl, H.-H., Westkämper, E.: Order Management in Transformable Business Structures-Basics and Concepts. Robotics and Computer-Integrated Manufacturing 17(6) (2001) 461-468
10. Abecker, A., Bernardi, A., Maus, H., Sintek, M., Wenzel, C.: Information Supply for Business Processes: Coupling Workflow with Document Analysis and Information Retrieval. Knowledge-Based Systems 13 (2000) 271-284
11. Kwan, M.M., Balasubramanian, P.: KnowledgeScope: Managing Knowledge in Context. Decision Support Systems 35 (2003) 467-486
12. Fensel, D., Motta, E., van Harmelen, F., Benjamins, V.R., Crubezy, M., Decker, S., Gaspari, M., Groenboom, R., Grosso, W., Musen, M., Plaza, E., Schreiber, G., Studer, R., Wielinga, B.: The Unified Problem-Solving Method Development Language UPML. Knowledge and Information Systems 5 (2003) 83-131
13. Smith, M.K., Welty, C., McGuinness, D.L. (eds.): OWL Web Ontology Language Guide. W3C Recommendation 10 February 2004. http://www.w3.org/TR/owl-guide
14. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21 May 2004. http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/
15. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge Engineering: Principles and Methods. Data and Knowledge Engineering 25 (1998) 161-197
16. Protégé. http://protege.stanford.edu/
17. Horridge, M., Knublauch, H., Rector, A., Stevens, R., Wroe, C.: A Practical Guide to Building OWL Ontologies with The Protégé-OWL Plugin and CO-ODE Tools. 2004. http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf
18. OWLViz – A visualisation plugin for the Protégé OWL Plugin. http://www.co-ode.org/ downloads/ owlviz/OWLVizGuide.pdf
19. OWL Web Ontology Language for Services (OWL-S). http://www.w3.org/ Submission/ 2004/07/
20. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual. 2nd edn. Addison-Wesley, Boston Massachusetts (2005)
21. Friedman-Hill, E.J.: JESS, The Rule Engine for the Java Platform. http://www. jessrules. com/jess/docs/70/

A Distributed IR Model Based on Semantic Web

Pei-guang Lin[1,2], Xiao-zhong Fan[2], Ru-zhi Xu[1], and Hai-yan Kang[3]

[1] School of Computer & Information Engineering, Shandong University of Finance,
Jinan 250014, P.R. China
llpwgh@bit.edu.cn

[2] School of Computer Science and Technology, Beijing Institute of Technology,
Beijing 100081, P.R. China

[3] Department of Computer Information System, Beijing Information Science &
Technology University, Beijing 100101, P.R. China

Abstract. Most of the current information retrieval methods are mainly based on keywords matching, and they can not understand the meaning of the keywords. Though some researches have proposed the methods based on domain ontology, it is difficult to implement the general IR because different ontology bases are heterogeneous. Aiming at the questions above, a distributed IR model based on semantic web services (D-IRSW) is proposed. This model puts different Semantic Retrieval Service (SRS) on special ontology base and the results returned by SRS are processed by Semantic Retrieval Service Engine (SRSE). Experiment shows that this model can improve the precision and recall of IR obviously.

1 Introduction

Tim Berners-Lee proposed the concept of Semantic Web in 1998, whose target is to develop a series of languages and techniques which can express semantic information and can be understood and processed by computers. The implementation of the semantic web is based on ontology which is defined as a formal, explicit specification of a shared conceptualization by Studer and et al [1]. The ontology has perfect concept hierarchy and supports logic inferences, so it is widely used in IR, especially in knowledge based IR. There are three famous projects, (Onto) 2 Agent, Ontobroker and SKC (Scalable Knowledge Composition, which use ontology in IR and delegate three aspects. The main objective of (Onto) 2Agent [2] is to help user to find the required ontology on WWW; that of Ontobroker [3] is to help user to find the pages on WWW which include the content he really needs; and that of SKC [4] is to resolve the problems of heterogeneousness of information system and to implement inter-operation of heterogeneous autonomy system. This paper focuses on the second aspect and D-IRSW, a Distributed Information Retrieval model based on Semantic Web, is proposed in this paper.

The next section describes the details of this model. The third section gives an experiment of the model and evaluates the performance. Then it comes to a conclusion in the last section.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 725–731, 2006.

© Springer-Verlag Berlin Heidelberg 2006

2 The Model of D-IRSW

This model is divided into 3 parts: Semantic Retrieval Service (SRS), Semantic Retrieval Service Engine (SRSE) and the definition of their interfaces. Fig. 1 shows the architecture of this model.

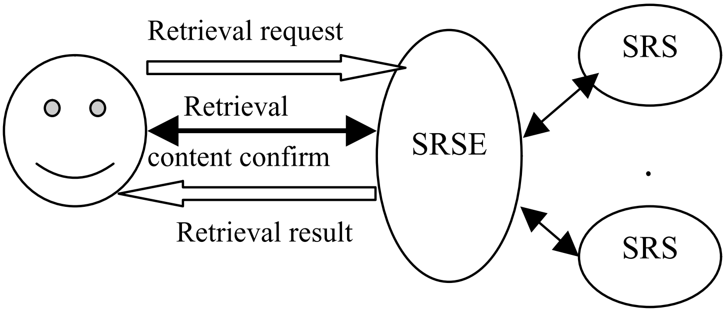


Fig. 1. D-IRSW System Metod

2.1 The Definition of the Interfaces

Three primitives are defined by the model so that SRSE can call the SRS effectively. They are "Ask", "Query", and "Answer" primitives. In order to agree with the standard of semantic web, the three primitives are defined by OWL. The function of "ask" primitive is to get some related information from SRS by SRSE, including, 1) whether supplying the service or not; 2) the classification of the service if existing. We can judge whether the service existing or not by the former and the later gives us the contents of the current web site so that SRSE can decrease the retrieval scope; 3) the query of the synonymous terminologies. The SRSE submits the retrieval requests to the SRS when it knows the semantic web site supplies SRS and the SRS belongs to the classifications that user specifies. The submission of the retrieval requests is implemented using "query" primitive. The retrieval requests may include more than one keyword and these keywords may have different relations, that is, there are different Boolean operations on these keywords. The OWL can express these easily. The "answer" primitive includes "answer-ask" primitive and "answer-retrieval" one. The former also includes "answer-classification" one and "answer-synonymy" one. The "answer-ask" responds the SRSE's "ask" primitive. The SRS returns its retrieval result using the "answer-retrieval" primitive. The result includes two parts: answer collection and termination-token. The former contains the retrieval results, including related resources (web pages), resource related degree and related keywords; the later includes End (to return all), None (to return nothing) and Rejected (not to process the requests).

2.2 Semantic Retrieval Service (SRS)

This module is the basic part of the model and is a kind of semantic web service distributed on different ontology base, whose function is to retrieve the local ontology base and return the result to SRSE using the "answer - retrieval" primitive. Firstly, this module gets the retrieval request sent by SRSE using the "retrieval" primitive; Secondly, it translates the retrieval request into local retrieval request (OWL-QL) and retrieves the ontology base; Finally, the retrieval result is returned to SRSE. Fig. 2 shows the architecture of this module.

2.2.1 Terminology Match

To construct rational ontology base is the precondition of constructing and retrieving semantic web. Ontology is the description about static concept models of certain domains, using acknowledged terminology set and their relations of these terms to reflect knowledge and knowledge structure. At present, the terminology set accepted by the domain does not exist, so the same ontology in a domain may be described in different way. That is to say, it is necessary to adopt the distributed semantic web service to complete the retrieval.

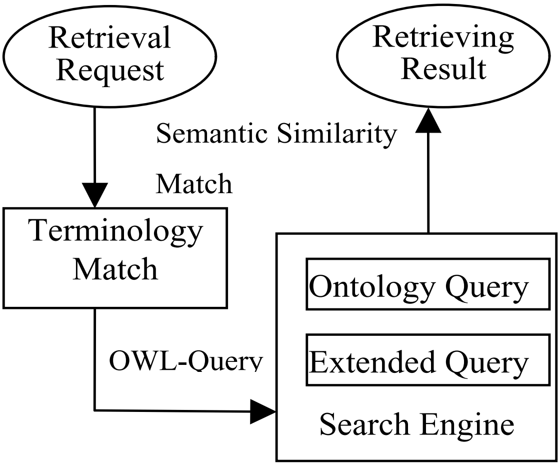


Fig. 2. Architecture of SRS

SRS receives the "query" primitive sent by SRSE, and gets the keywords and the relations between keywords. After the keywords are compared with the local terminologies, the ontology and its properties and the relations that match best are returned. To compute the semantic similarity between the terminologies and keywords, we build a synonym table related with each terminology, and then we can get it by computing the similarity of the synonym and keywords. In this way, we map the keywords to the terminologies of ontology base. We introduce the terminology matching process using the example of "printer" ontology in reference [5]. In order to relate the web page with each ontology, we add the "related-page" property to each one, so the user can view detailed information about the current ontology. When the ontology base is created for a semantic web, the terminologies can be extracted from it and the synonym table can be created to process word. After that, we can compute the similarity of keywords and terminology (including synonym table). If there is more than one synonym, we select the one with maximum similarity. Using "Literal Similarity (LS)" as calculating method, we define the same morpheme similarity "alpha" value to 0.722, the same morpheme location relevant similarity "beta" value to 0.278, and we set the threshold value as 0.4 and if the similarity is lower than that, we ignore the terminology.

2.2.2 Search Engine

We map the keywords to terminologies by matching operation and these terminologies delegate the name, property and relation of the ontology. Then we can complete the retrieval operation by constructing the retrieval clause using OWL-QL. When retrieving, we discuss several cases as follows: 1) If the matching terminologies only contain the name, property or relation of the ontology, the matched ontologies, the ontologies including the defined property, or the ones including the defined relation will be returned. 2) If the matching terminologies contain the name and property of the same ontology, the matched ontology will be returned; otherwise, this request will be processed as 1); 3) If the matching terminologies contain the name and relation of the same ontology, the ontology will be returned; otherwise, this request will be processed as 1); 4) If the matching terminologies contain the property and relation of the same ontology, the ontologies including the property and related properties will be returned; otherwise, this request will be processed as 1); 5) If the matching terminologies contain the name, property and relation of the same ontology, the ontology will be returned; otherwise, this request will be processed as 1); After finishing the retrieval, SRS returns the related ontologies including related web pages (resources). Then we can implement the extended retrieval by using the relation of property (e.g. subPropertyOf) and ontology (e.g. subClassOf, equivalentClass). Because the relations are predefined, we define b as the weight of each relation and we set b=1 when the ontology is retrieved directly. So,g, the similarity between the resources and keywords is defined as g=a*b. Finally, SRS orders the result by g and returns them through "answer - query" primitive.

2.3 SRS Engine (SRSE)

The SRSE works as following steps:1) After getting the retrieval requests, SRSE begins to retrieve data by calling related distributed SRSs;2) When receiving the retrieved results, the resources are filtered and sorted by SRSE;3) Finally, the processed results are sent to the user. Fig. 3 shows the architecture of SRSE.

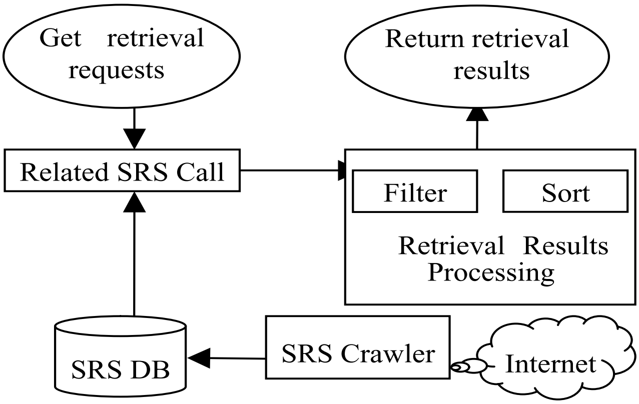


Fig. 3. Architecture of SRS Engine

2.3.1 SRS Database (SRS DB) and SRS Calls (SRSC)

In order to call SRS conveniently, an SRS DB is maintained on server, including their URL and catalog. We classify the SRS by Chinese Library Classification (CLC). After a user inputs the retrieval request and specifies the content classification, SRSE gets all the SRS of the specified classification and all its descendants, and the weight of each SRS, W, is set to 1. Then, SRSE gets the ancestors of the SRS and while the level rises, the weight will automatically changes to the reciprocal of the number of child classes of its parent level. We give a threshold q, and when q is smaller than 0.3, SRSE stops tracing back. For example, in CLC, "TP3" has eleven subclasses, so its weight changes to 1/11 of its origin value which is equal to the maximum weight of all its subclasses.

2.3.2 SRS Crawler

In order to keep the SRS available, the SRS DB is updated by special SRS crawler regularly which can get new SRS, delete invalid SRS and update their classification by using "ask" primitive. After retrieving, SRSE selects top Ni results from each SRS's result as the initial set. The value of Ni is related to the size of the result returned by SRS and the weight of SRS. Suppose Si is the SRS I, Ri is the result return by Si and Wi is the weight of Si, we get:

$$Ni = C \times |Ri| \times Wi / \sum_{i=1}^{M} Wi. \tag{1}$$

In which, the C is constant and can be assigned the value 0.1, 0.5 etc and $|Ri|$ is the cardinal number of set Ri. Wi is generated dynamically based on the matching degree between keywords and related catalogues when retrieving, which is used to express the reliable degree. After doing this, SRSE gets rid of repetition and sorts the result, then the processed result is returned to the user.

3 Experiment and Analyses

We evaluated this model by comparing to keyword based search engine against a database of 400 web pages about computer. We use the ontology base about printer in reference [5] and add link to each page. Then we count the number of pages about following keywords: ”printer price”, ”HP printer” and ”Laser printer” and get their average precision and recall.

Figure 4 shows the results of our model and the traditional SE.

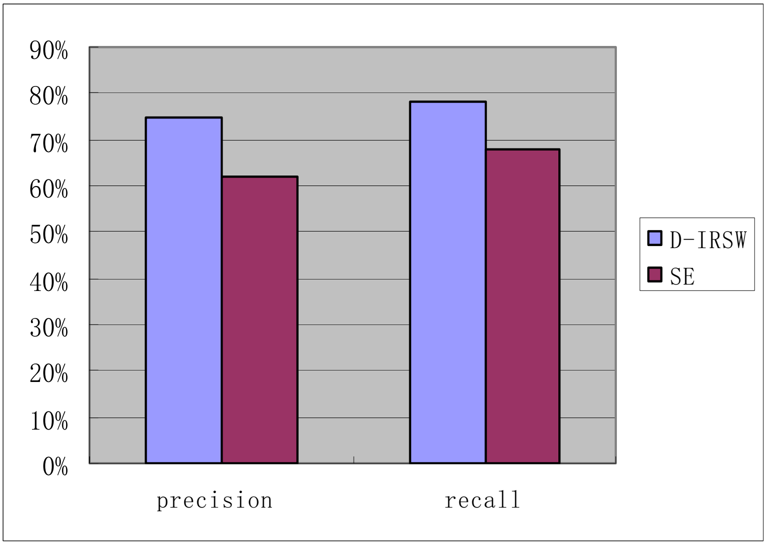


Fig. 4. Comparison of D-IRSW with Conventional IR

Following the example, we compare the D-IRSW model with traditional search engine: 1) About the retrieval result, the traditional search engine returns the web page’s abstract and URL, but our new model returns the structural ontology information and URL. As a result, a user can view the information more conveniently. Because of using a concise ontology base as web page’s data source, the results will not lose any useful information. It can also improve the precision and include less redundant information. 2) Because traditional search engine bases on keywords matching, some pages in higher position of the results may not include any relative information. Instead, this new model can directly return the structural price information in need. 3) This new model is very easy to realize a general IR system. The reasons are: a) The SRS is distributed and implemented on special ontology base, so it does not care about the heterogeneousness of the ontologies. b) When SRSE integrates all the results, the ontologies are sorted by the related degree of resources. And the repeated one is deleted by the ontology’s resource ID.

4 Conclusions

Because the retrieval model, D-IRSW, converts the user's retrieval requests and the documents to semantic information that can be understood by computer, it will be very beneficial to the implementation of computer reference. It will also improve the recall and precision because of including rich semantic information. The retrieval operation is implemented on different ontology base based on separated and distributed semantic retrieval service. As a result, not only the disadvantage of ontology heterogeneousness is compensated, but a personalized retrieval solution is delivered for different contents of the websites which provides higher quality of services. Semantic web is the next generation of the internet and this research has made beneficial contributions to this interesting topic - information retrieval on semantic web.

References

[1] Studer R, Benjamins V R, Fensel D. Knowledge Engineering, Principles and Methods. Data and Knowledge Engineering, **25**(1998),(1-2)161-197

[2] Arpirez J, Perez A G, Lozano A, et al. (Onto) 2 agent: An Ontology – based WWW Broker to Select Ontologies. In Proc. of the Workshop on Application of Ontologies and Problem – Solving Methods, UK, (1998)16 - 24

[3] Ontobroker. http://ontobroker.aifb.uni-karlsruhe.de

[4] SKC. http://www-db.stanford.edu/SKC

[5] Grigoris Antoniou, Frank van Harmelen. A Semantic Web Primer [M]. [London]The MIT Press, 2004 McIlraith S A, Son T C, Zeng H. Semantic Web Services. IEEE Intelligent System, **3/4**(2001)46 - 53

Experimental Study of Semantic Contents Mining on Intra-university Enterprise Contents Management System for Knowledge Sharing

Keiko Shimazu[1], Isao Saito[1], and Koichi Furukawa[2]

[1] Research Institute for Digital Media and Content, Keio University, with a grant from the
Ministry of Education, West Annex, 2-17-22 Mita, Minato-ku, Tokyo 108-0073 Japan
{shimazu, 130s}@dmc.keio.ac.jp
[2] Graduate School of Media and Governance, Keio University
Endo 5322, Fujisawa-City, Kanagawa, 252-8520, Japan
furukawa@sfc.keio.ac.jp

Abstract. We developed an Enterprise Contents Management System for an academic domain. The main feature of this system is its function for focusing searches in Web documents, utilizing human names and locations appearing in the documents as the search context. To realize this function, we adopted a standard text-mining algorithm for extracting proper nouns. We conducted an experimental study of this system against the existing digital contents of our university, and succeeded in efficiently obtaining suitable contents along the given contexts, which were obtained through previous searches. This experiment also suggested that our approach solves the general problem of finding an appropriate set of key words in a Web search. By performing this experiment, we confirmed that context mining is one of the most important technologies to be further developed in our effort to promote knowledge circulation through digital contents.

1 Introduction

Recently experimental studies in knowledge discovery or innovative thinking have become more popular than ever. When these studies are performed, there needs to be information sharing network system working on a broad scale. In the industrial field, this system is called an Enterprise Contents Management (ECM) system. Lately commercially available versions of ECM systems have been developed. If we focus on the issue of knowledge sharing, we need to realize the functions of handling background knowledge and extracting only the information that meets the user's demands or intentions. However, existing ECM systems mainly have only those modules for maintaining a repository of digital contents and searching contents by identifying keywords. Therefore, finding the right contents by entering the proper keywords is still a user intensive effort, as is researching background knowledge of those contents. The heuristic functions that can assist human intellectual endeavors have not yet been implemented on existing ECM systems.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 732 – 738, 2006.
© Springer-Verlag Berlin Heidelberg 2006

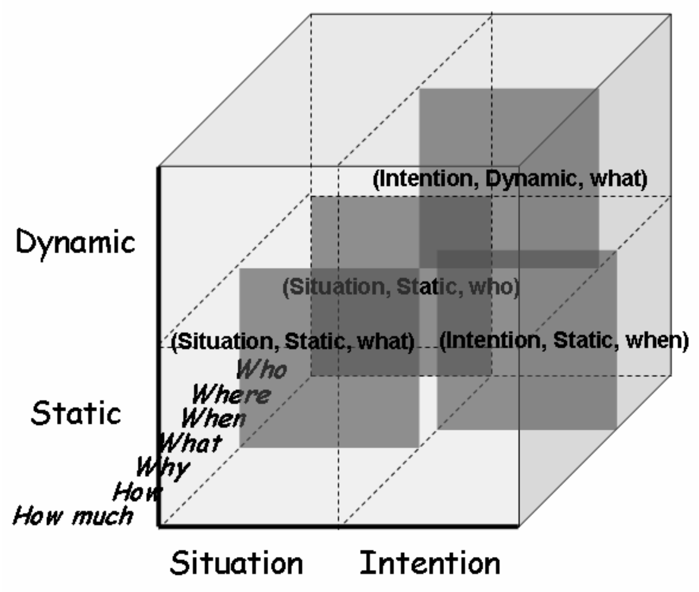


Fig. 1. Our 3D model

The remainder of this paper is structured as follows; section 2 summarizes work related to our experimental study. Section 3 clarifies the objective of our experimental study. Section 4 explains our ECM system, while section 5 reports on its examination and subsequent evaluation. Section 6 discusses on the results of examination. Finally, section 7 presents a conclusion.

2 Our 3D Model

In general, when researchers impart their knowledge to others in the same field by means of digital contents, they add additional information in metadata to clarify the necessary context (i.e. background knowledge) in order to add additional information in metadata. In the field of sociology, 5W1H, who, what, why, where, why and how, is the standard solution for this problem. It is thought that if people use this solution, they can thoroughly communicate essential knowledge. In the field of marketing, in order to assimilate the concept of "circulation", the solution 5W1H + H was proposed in which the added "H" stands for "how much". Because the focus of our discussion is also the environment of contents circulation, 5W1H + H is employed for abstracting metadata. On the other hand, neither the field of sociology nor the field of marketing has the concept of "storage". Their concepts focus on the flow of information or articles of trade. In our field, storing contents is an important issue. When contents are stored on a web server once, they are available to any one. Even if contents are stored for a unique purpose by their creator, they might be utilized for another application. The value of contents and the demographic of the users also changes as time passes. For handling this concept, the idea of [1] is employed to be integrated into the idea of 5W1H + H. One of the dimensions is 5W1H + H, which is “who, what, when, where, why, how or how much”. The second dimension is Static or Dynamic. The third dimension is Situation or Intention. To be more precise, all items of metadata which are placed in XML schemata explicitly are abstracted by mapping them onto our 3-dimensional space (Fig 1). The extracted metadata are mapped onto chosen areas of the space, the structure of which is ({Situation, Intention}, {Static, Dynamic}, {who, what, when, where, why, how, how much}).

3 DMC Network System

3.1 Mapping onto Our 3D Model

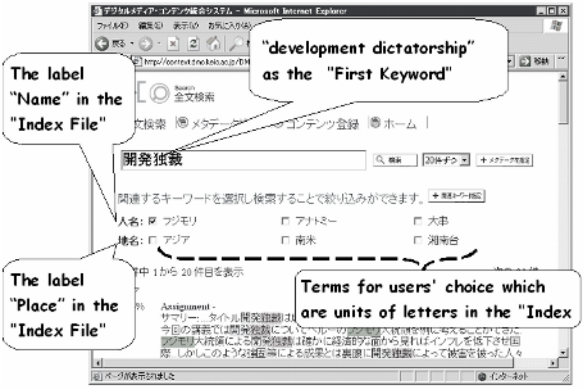
As mentioned above, information gathered by our crawler was parsed, tokenized and annotated, resulting in what we call, expediently, an annotated file. On the other hand, the three axes of our 3D model are structured in the following way: the X-axis, which has the two values of “Situation” and “Intention,” the Y-axis, which has the two values of “Static” and “Dynamic,” and the Z-axis, which has the seven values of “Who”, “Where”, “When”, “What”, “Why”, “How” and “How much.” In more mathematical terms, the values are mapped onto the axes (x, y, z) on our 3D model, where {x|x= “Situation”, “Intention”}, {y|y= “Static”, “Dynamic”}, {z|z=“Who”, “Where”, “When”, “What”, “Why”, “How”, “How much”}. Therefore, to visually map the terms onto our 3D model is to put them onto any of 28 squares (Fig 1). In this process, our system refers to the mapping list. Additional identification information regarding each unit of letters is labeled on each token. Through this function, the terms “撰者’ (pronounced Sen-Ja) and “composers” are mapped onto the same square. Using this framework, our system understands that the contents of both fields have the same metadata schema labeled “Name”, because all of these all values are mapped onto the same square of (Situation, Static, Who). Additionally, this information is added to the annotated file. This information is used at the time of the contents search refinement described in detail in the following chapter.

3.2 The System Flow as Focused in the Users’ Operations

A data gathered by the crawler is converted in the first step. In this paper, this converted data is named the “index file.” When the user performs the contents-search on the home page of the website of the DMC network system, the “index file” is used to choose the target contents. As a result, our system displays[1] the target contents which have the same keywords as the user’s input[2]. In this paper, the keyword(s) for the contents-search will be referred to as the “first keywords” and this output of the contents is named the “first set of contents.” Depending on the user’s request, our system will show other keywords for search refinement. These terms are selected from the “first set of contents”; they are units of letters whose schema is labeled “Name” and

[1] Our DMC network system will not display all the contents following 500th, because it is impossible to handle large volume of them.

[2] At the time of output, mathematical formula about the keyword(s) is executed for each content. It is unique, having been developed only for this platform system.



デジタルメディア・コンテンツ統合システム - Microsoft Internet Explorer
全文検索
メタデータ
コンテンツ登録
ホーム
検索
20件ずつ
人名:
地名:

開発独裁

関連するキーワードを選択し検索することで絞り込みができます。

人名: ☑ フジモリ □ アナトミー □ 大串

地名: □ アジア □ 南米 □ 湘南台

中 1から 20 件目を表示

Assignment -
サマリー:...タイトル開発独裁は
今回の講義では開発独裁についてペルーのフジモリ大統領を例に考えることができた。
フジモリ大統領による開発独裁は確かに経済的な面から見ればインフレを低下させ国
際...しかしこのような強圧等による成果とは裏腹に開発独裁によって被害を被った人々

ページが表示されました

"development dictatorship" as the "First Keyword"

The label "Name" in the "Index File"

The label "Place" in the "Index File"

Terms for users' choice which are units of letters in the "Index

Fig. 2. actual image on a display

"Place" in the "index file." The top three terms of "Name" and "Place" are displayed on the user interface, depending on their entropy values. When the user selects one or more of the choices, our system searches the target contents, which contain the "first keywords" and also the keywords from the choices. Finally our system displays the search result.

4 Experimental Study on Practical Use

Our DMC network system's crawler gathered about sixteen thousand URLs from Keio Networks[3]. Our experimental study on practical use was performed from 8th Jan. 2006 to 2nd Feb.[4] excepting Saturdays and Sundays, by 7 experimenters who were research assistants. As mentioned before, our DMC system was built on the actual intra-university networks; therefore the crawler of our system gathered actual data on its web servers and databases. It could be that there are an enormous number of keywords combinations any content-search. Therefore, our experimenters selected the keywords in a random manner and without a plan. Since it was assumed that most contents and metadata were written in Japanese, only Japanese keywords were used. In the following chapters, for the purpose of presenting our study more clearly, significant results are translated into English. (Fig2 shows a example of actual outputs.)

4.1 Content-Search with Conventional Procedure

The content-search procedure known in general was experimented. The "first keyword" was "data mining." As the "first set of contents", over 500 contents were displayed[5]. As the multiple keywords, "inductive inference", "deductive inference" and

[3] keio ac.jp and keio.edu

[4] Because the target content on our networks has been changing everyday, it is not necessarily correspond to the present outputs of the contents-search on our system.

[5] Our DMC network system will not display all the contents following 500th, because it is impossible to handle large volume of them.

"entropy" were entered following the "first keyword", "data mining." The number of content as the result of content-search was reduced to 169 from over 500 as the "first set of result." On the other hand, at the case of the using "Jun Murai" as the "first keyword", even "WIDE" was added as a combined keyword, the number of contents as a result was still over 500.

4.2 Utilizing of Metadata on Our 3D Model

4.2.1 Utilizing (Situation, Static, Who)

The menu of "Name" has the tree selectable words which were values of metadata that schema name were abstracted to be mapped on to the square of (Situation, Static, Who) on our 3D model. In our experimental study, over 500 contents displayed for the "first keyword", "knowledge share." According to our experimenter's request, our system displayed 3 names for "Name." After selecting on of them, "Yasushi Kiyoki", the number of content was reduced to 105 and list of words was changed to display new words. Furthermore, as other word "Naofumi Yoshida" was selected on the new list, the number of the content was reduced to 22.

4.2.2 Utilizing (Situation, Static, Who) and (Situation, Static, Where)

The menu of "Place" has the tree selectable words which were values of metadata that schema name were abstracted to be mapped on to the square of (Situation, Static, Where) on our 3D model. In our experimental study, 450 contents displayed for the "first keyword", "development dictatorship." According to our experimenter's request, our system displayed 3 names for each "Name" and "Place." As the "Fujimori" was selected on the list of the former, the number of content was reduced to 102. On this situation, as the "South America" was selected on the list of the "Place", the number of content was reduced to 21.

On the other hand, after displaying 450 contents, as the "Paku" was selected on the list of the "Name", the number of content was reduced to 105. On this situation, as the "Soul" was selected on the list of the "Place", the number of content was reduced to 4.

4.2.3 Combining Conventional Procedure and Ours

In our experimental study, 163 contents displayed for the "first keywords", "stock price crash." As the "Bush" was selected on the list of the former and "Baghdad" was for the latter, the number of content was reduced to 70.

On the other hand, after displaying 163 contents, as the "Koizumu" was selected on the list of the "Name" and "Asia" was for the "Place", the number of content was reduced to 154.

5 Discussions

5.1 Contents-Search Refinement by Combination of Keywords

The former case of Chapter 5.1 is the example of success in search refinement using the combination of keywords. On the other hand, the latter is the example of failure. The reason of this difference between these results is that in the former case the experimenter had a thorough knowledge of data mining which had used as the "first keyword", in the latter case the experimenter didn't. Therefore, the former one was easy to find the keywords for refinement of contents-search. In general, this situation is common occurrence. If users want to succeed in striking the target contents, the users are need to be the experts of the field to prepare the suitable keywords. It's almost certain that the efficient contents-search is depends on users' competency of keywords selection and experiment. Therefore the quality of result depends on largely on users' experiment.

5.2 Contents-Search Refinement

In the cases of Chapter 5.2.1, 5.2.2 and 5.2.3, the contents-search refinement was performed successfully; even the experimenters didn't enter the keywords followed by the "first keyword." In the first case the number of the "first set of contents" using the "first keyword" was over 500. If users are not the experts of the field, it is near impossibility to perform the contents-search refinement using appropriate prepared by the users endeavor. Therefore the users need select the contents randomly and open and read them to affirm their appropriateness. This is physically impossible, because that in the case of those 3 chapters, the over 450 contents were displayed.

In those cases, our DMC network system showed the term choices of "Name" and "Place." The experimenters selected one or more among them to refine the contents-search effectively.

5.3 Extracting the Context

In the cases of 3 chapters mentioned above, our system displayed the choices of the terms. On each occasion of the users' selection of terms, our DMC network system displays the new set of the terms depending on the new result of the entropy computation. Therefore, on each occasion of the users' selection of terms, any context is gradually generated because the context is produced by connected contents. For instance, in the case of Chapter 5.2.1 the experimenter generated the context for the corroborators of knowledge share. In the case of Chapter 5.2.2, the experimenter generated the context for the two actual examples of development dictatorship. The examples present the context that "Who" did it at "Where." Also in the case of Chapter 5.2.3, the experimenter generated the context for the two actual examples of stock price crash. The examples present the context that "Who's" speech and behavior about/at "Where." As remarked above, in this experimental study, our DMC system supported users to generate the contexts which created by digital contents.

5.4 More Challenges

The contents-search was succeeded depending on the particular contexts, in the chases of Chapter 5.2.1, 5.2.2 and 5.2.3. On the other hand, not enough effective reduction of the number of contents from contents-search was succeeded in Chapter 5.2.3. Our DMC network system searches the target contents in the "index file" on each occasion of users' selection of the terms for contents-search refinement. This algorithm causes potentially not enough effective reduction. Our next challenges are to extract the semantic concepts automatically. The concepts are represented in the 28 parts of our 3D model. Those are used for the abstraction refinement in our DMC network system.

6 Conclusion

We proposed 3D model for contents sharing across various disciplines. We implemented a breakthrough application program employing text mining technology on our intra-university enterprise contents management system, which was built for this experimental study and was directed to all-contents-campus-wide.

Reference

1. Kiyoki, Y., ``A Metadatabase System for Semantic Image Search by a Mathematical Model of Meaning, '' ACM SIGMOD Record, vol. 23, no. 4, pp. 34-41, 1994.

Semantic Autocompletion

Eero Hyvönen and Eetu Mäkelä

Semantic Computing Research Group (SeCo)
Helsinki University of Technology (TKK), Laboratory of Media Technology
University of Helsinki, Department of Computer Science
FirstName.LastName@tkk.fi
http://www.seco.tkk.fi/

Abstract. This paper generalizes the idea of traditional syntactic text autocompletion onto the semantic level. The idea is to autocomplete typed text into ontological categories instead of words in a vocabulary. The idea has been implemented and its application for semantic indexing and content-based information retrieval in multi-facet search is proposed. Four operational semantic portals on the web using the implementation are presented as application cases.

1 Introduction

The idea of *autocompletion*[1] is to predict what the user is typing in, and to complete the work automatically. The benefits of this simple idea are manyfold: First, the computer helps the user in memorizing the right vocabulary used. Second, typing errors in the input can be minimized. Third, autocompletion speeds up the interaction. A side effect of the idea is that it encourages the usage of long descriptive names and commands that are more understandable to the users. An idea related to autocompletion is *autoreplace*, where the idea is to use predefined abbreviations in typing and the system automatically replaces these with full-blown strings.

In order to make the prediction right and as early as possible, the underlying vocabulary must be known, be limited, and the words in the lexicon should differ from each other in terms of the leading characters. These conditions hold in many applications, such as operating system shells, email programs, browsers, etc.

Autocompletion is used, e.g., in Microsoft's Intellisense feature of the Visual Studio, where the idea is applied to source code editing. Here a pop-up menu is used to show the programmer possible autocompleted forms. This is useful when it is difficult to remember or type in, e.g., the names of the methods of a particular class at hand. A widely used application of autocompletion is the predictive text entry system in mobile phones [1,2] commonly known as T9, where only a limited number of keys are available instead of the full QWERTY keyboard. By associating each key with a set of letters (e.g. '1' with a, b, and c)

[1] See e.g. http://en.wikipedia.org/wiki/Autocompletion

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 739–751, 2006.
© Springer-Verlag Berlin Heidelberg 2006

and by completing single keypresses automatically based on a dictionary, input typing can be speeded up significantly e.g. in text messaging.

Autocompletion can be done *by request* or *on-the-fly*. In Linux/Unix and DOS operating systems, for example, the command line is completed—or possible continuations are shown—after a hit on the TAB-key. The on-the-fly-approach is used e.g. in browsers and email-systems: the text typed in is completed into matching URLs or email addresses that have been used before, or are stored in an address book. A nice recent application of autocompletion on-the-fly on the web is the beta version of Google Suggest[2] that completes input text into feasible search keywords.

Traditional autocompletion is based on matching input strings with a list of usable words in a vocabulary. This paper generalizes this approach onto the semantic level. The idea is to complete user written text not only into similar words, but into matching ontological concepts whose labels may not be related to the input on the literal level. For example, the typed input 'preside...' could be autocompleted into 'George W. Bush' since George W. Bush is an instance of the class president. It is also possible to complete the input text into the different homonymous meanings (concepts) of the input, and into the different semantic roles in which the concepts are used. This possibility provides the end-user not only with a semantic matching service but can be used to disambiguate the meanings and thematic roles in which the concepts are used. To continue the example above, input 'preside...' could be autocompleted into 'George W. Bush (as an author)' or 'George W. Bush (as a document subject)'. By providing the autocompleted choices to the end-user, the right interpretation can be disambiguated and, for example, search be performed with the right meaning.

In the following, this idea to be called *semantic autocompletion* is first discussed as a means for semantic information retrieval, and some of its different forms are identified. After this, implementation of the idea in the OntoViews framework [3] is presented, and application in three semantic portals for concept-based information retrieval and in semantic indexing is exemplified.

2 From Syntactic to Semantic Completion

We consider the idea of semantic autocompletion in information retrieval, especially, in multi-facet search [4,5,6,7]. Multi-facet search is a generalized form of the traditional single-facet search paradigm. Examples of single-facet systems include Yahoo!, Open Directory Project[3], and many traditional web portals. In multi-facet search, content is organized and retrieved using multiple hierarchical structures at the same time, instead of just one like in single-facet search.

2.1 Autocompletion in Multi-facet Search

In multi-facet systems the data has been indexed using keywords from a set of hierarchical orthogonal facet categories. For example, in [5] the facet categories

[2] http://www.google.com/webhp?complete=1&hl=en

[3] http://dmoz.org/

of the Art and Architecture Thesaurus AAT[4] are used as subject terms. The location facet divides the earth into continents (Africa, Antarctica, Asia, ...), each continent consists of countries, and each country is divided further into counties, cities, etc. The material facet is a classification hierarchy of materials used or depicted in the collection items. The search objects are classified along facets based on the keywords used in annotating the collection items. The user selects categories from different facets and the search result is the intersection of the items belonging to the selected categories. By selecting a supercategory, all hits related to its subcategories (recursively) are returned, too. Let mapping $m : S \rightarrow C$ map each search item $s \in S$, where S is the set of search times, to the set of facet categories C. Then the hit set H corresponding to selected search categories $c_1, ..., c_n$ is $H = \{s|c_i \in m(s), i = 1, ..., n\}$.

In traditional multi-facet search, the keywords are strings as usual in keyword search. In [6] multi-facet search is extended with semantic web ontology techniques and reasoning. The idea is to replace keywords with ontological resources in indexing and then determine the mapping m between search categories and search items using logical mapping rules. In this way, multi-facet search can be generalized onto a semantic level where the mapping between facets and search items can be based on semantic relations and not only on simple keyword match. For example, in [8] the category 'Nokia' as a company in an actor facet is mapped onto different search items than 'Nokia' as a city in Finland in a location facet.

Semantic autocompletion in multi-facet search can be defined as a function $f : text \rightarrow < C, H >$ that maps an input string $t \in text$ onto a set of search categories of the facets C and the corresponding search item hits H in the data set. The hits are based on the different semantic meanings of the input. For example, if the user types in the word 'bank', this could be completed into categories 'river bank' and 'bank (financial)' and the result set includes an union of both geographical and organizational hits.

The input may consist of several partly written keywords that correspond to category selections. For example, 'Finl presid' could mean that the user searches information about categories 'Finland' and 'president', e.g., about the presidents of Finland. The categories C and hits H matching the input should, in the user's view, match in meaning with the intended meanings of *text*. For example, input 'Scandin...' may match the category 'Nordic countries'. Notice that here 'Scandinavia' and 'Nordic countries' do not share substrings as required in traditional autocompletion. In our case, autocompletion is occuring on the semantic level in the user's mind, and is implemented using the underlying ontological structures.

Autocompleting an input string into facet categories can be based on several principles. In below, some forms of autocompletion are discussed.

2.2 Autocompletion Based on Equivalence Relations

This form of autocompletion deals with the problems of lexical variants, synonymy, polysemy, and homonymy. Lexical variants and synonyms are alternative terms

[4] http://www.getty.edu/research/conducting_research/vocabularies/aat/about.html

that correspond to the same ontological concept. For example, 'NYC', 'New York City', and 'Big Apple' refer to the same city. Semantic autocompletion can provide a service, where typing in any of the terms is completed into the same concept, denoted by its preferred term, here 'New York City'.

This kind of autocompletion can be enabled to some extent by listing alternative and preferred labels for concepts. If the input matches any of these, the corresponding concept is selected, and the preferred label is shown to the user. However, in morphologically rich languages, such as a Finnish, listing all morphological variants as explicit alternatives may not be feasible, and dynamic morphological analysis may be needed as a part of autocompletion before ontological matching. For example, the genitive plural for of the Finnish word 'yö' (night) is 'öiden', a literal quite different from the nominative form.

In polysemy, a single term has different but related meanings (e.g., 'arrow head' and 'human head'); in homonymy the meanings are totally different (e.g., 'river bank' and 'blood bank'). In both cases, the meaning cannot be disambiguated based on the user's shorthand input ('head' or 'bank'). The same happens when the user's partial input can be completed in different ways (e.g., 'New' $\mapsto$ 'New York' or 'New' $\mapsto$ 'New Year'. In these cases the autocompletion function can provide the user with a list of possible choices from which to disambiguate.

One problem in determining the equivalence between input text and categories is how to deal with phrasal concept labels, such as 'broadband integrated services digital network'. Here, the categories can be matched against all permutations, and only the combinations leading to actual hits returned, so that for example the search can return the two-category combination 'Integrated Services + Digital Network (11 hits)' as a reasonable autocompletion, while the two-category combination 'Broadband Integration + Digital Services (0 hits)' is left out. In such complex multi-word labels, words may also appear in morphologically conjugated forms, which makes pattern matching more difficult, again possibly requiring morphological analysis as a pre-step. On the user interface level, one must also remember that particularly for compound words the matching part may not necessarily begin the input string, so that the prefix matching is not sufficient, but the whole string needs to be scanned for matches.

In multilingual autocompletion the keywords can be expressed in different languages and be matched on the same concept. This facilitates multilingual search even when the actual data is available or has been indexed in one only language. For example, 'bank (financial)' $\mapsto$ 'pankki (Finnish)'.

A benefit of semantic autocompletion is that the ontological environment of the matched categories can be visualized in addition to the actual matches. By showing the category hierarchy leading to the matched concept, the user can easily understand the meaning of the different completions. Furthermore, she can complete the text into the superclass or related concepts. For example, 'bank' $\mapsto$ 'financial institution > bank', where '>' indicates the subclass relation in the hierarchy.

2.3 Indirect Semantic Autocompletion

Semantic completion can be extended beyond equality to other semantic relations. The input string can be matched with not only the corresponding equivalent category, but with other related categories, too. For example, assume that you are looking for information about countries. By typing in 'EU' or 'US' semantic autocompletion could complete the text into a choice list of member countries of EU or states of the US, respectively, saving the effort of memorizing their names. Here the isPartOf-relation to is used for completing the text into neighboring ontological resources. However, in principle any arbitrarily complex relation could be used here, as long as its interpretation is intuitive and of use to the end-user.

2.4 Semantic Role Completion

An application of semantic autocompletion is *semantic role completion*. Here we not only match the input text with categories but also take into account the roles in which the categories are used. For example, the same city may be related with a museum collection artifact either as the place of manufacture or as the place of usage in the metadata. Depending on the choice, different result sets are obtained (unless all relevant items are both manufactured and used in the same place). Semantic autocompletion can provide the user with the possible choices to disambiguate.

2.5 Semantic Autocompletion Search

Semantic autocompletion can be combined seamlessly with semantic search. By completing the input string not only in related categories but also into the actual hits in the underlying data set, the user can actually see the hit list to narrow down as she types in text.

3 Application of Semantic Autocompletion

In the following we show by examples from various case studies, how the different forms of semantic autocompletion can be realized in practise in semantic information retrieval and indexing.

3.1 Semantic Category Search: Case MuseumFinland

Autocompletion can be used to disambiguate meanings in queries. This is useful especially if the content searched for has been annotated using correspondingly disambiguated concepts. An example of such a system is the semantic portal MuseumFinland[5] [7]. We have incorporated a version of semantic autocompletion into this application.

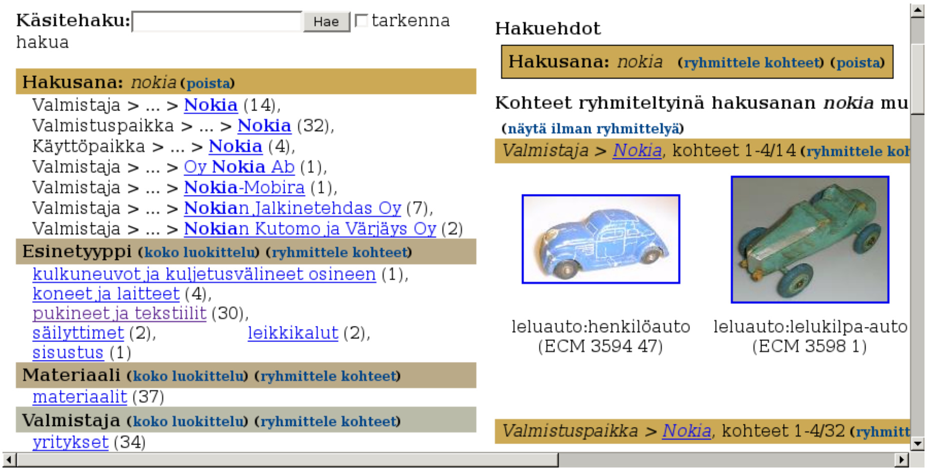
[5] http://www.museosuomi.fi

MuseumFinland integrates semantic autocompletion with multi-facet search. The search keywords are matched not only against the actual textual item descriptions, but also the labels and descriptions of the ontological categories by which they are annotated and organized into the view facets. As a result of semantic autocompletion, a new "dynamic facet" is created in the user interface. This facet contains all categories whose name or other configurable property value, such as alternative labels, match the keyword. Intuitively, the dynamic facet categories tell 1) the different interpretations of the keyword and 2) their roles with respect to the search items (here museum collection artifacts) in the metadata.

The result of a sample keyword search is shown in figure 1. Here, a search for input "nokia" has matched, for example, the following view categories:

- 'Nokia' as the telephone company and a manufacturer in the view Manufacturer ('Valmistaja' in the screenshot),
- 'Nokia' as a town in the view Place of Manufacture ('Valmistuspaikka'),
- 'Nokia' as a town in the view Place of Usage ('Käyttöpaikka'), and
- 'Nokia-Mobira', a predecessor of the telephone company, in the view Manufacturer.

By default, search is done by using the union of all possible interpretations. Search results are shown and classified according the possible choices on the right in the figure. However, the categories found can be used to constrain the multi-facet search further, with the distinction that selections from the dynamic facet replace selections in their corresponding facets and dismiss the dynamic facet. The right interpretation is selected by clicking on the corresponding link in the dynamic facet.



Käsitehaku: Hae □ tarkenna hakua

Hakusana: *nokia* (poista)
Valmistaja > ... > Nokia (14),
Valmistuspaikka > ... > Nokia (32),
Käyttöpaikka > ... > Nokia (4),
Valmistaja > ... > Oy Nokia Ab (1),
Valmistaja > ... > Nokia-Mobira (1),
Valmistaja > ... > Nokian Jalkinetehdas Oy (7),
Valmistaja > ... > Nokian Kutomo ja Värjäys Oy (2)

Esinetyyppi (koko luokittelu) (ryhmittele kohteet)
kulkuneuvot ja kuljetusvälineet osineen (1),
koneet ja laitteet (4),
pukineet ja tekstiilit (30),
säilyttimet (2), leikkikalut (2),
sisustus (1)

Materiaali (koko luokittelu) (ryhmittele kohteet)
materiaalit (37)

Valmistaja (koko luokittelu) (ryhmittele kohteet)
yritykset (34)

Hakuehdot

Hakusana: *nokia* (ryhmittele kohteet) (poista)

Kohteet ryhmiteltyinä hakusanan *nokia* mu
(näytä ilman ryhmittelyä)
Valmistaja > Nokia, kohteet 1-4/14 (ryhmittele koh

leluauto:henkilöauto (ECM 3594 47)
leluauto:lelukilpa-auto (ECM 3598 1)

Valmistuspaikka > Nokia, kohteet 1-4/32 (ryhmitt

Fig. 1. Using the keyword search for finding categories

In MuseumFinland, semantic autocompletion can be seen as search over a set of RDF(S) categories that correspond to classes in the underlying ontologies. At the same time, also hit lists of museum collection items are generated. This idea expanding queries over hierarchies has been applied also, e.g., in the Open Directory Project search engine. However, in our case the 9 category views have been projected, using a set of logical rules, from a set of 7 underlying ontologies in the system knowledge base. Matching is not straight-forward because of the projection, but indirect and more flexible. For example, in the search results of figure 1, the category 'Nokia' appears twice as a place (town). This is because the category can appear in the content of the portal in two different roles. Simply choosing e.g. the category 'Nokia (the place)' would not disambiguate the meaning sufficiently, since the same resource has the role of place of manufacture (Valmistuspaikka>...>Nokia) or place of usage (Käyttöpaikka>...>Nokia), or both, in the metadata of the museum artifacts. In the case of MuseumFinland, these roles can be disambiguated automatically by semantic autocompletion: the user can choose from a list of given options the correct role meaning of the keyword 'nokia' indicated by the subcategory path leading to it.

3.2 Semantic Autocompletion on the Fly: Case Orava

In MuseumFinland autocompletion is done on request, i.e., after pushing the search button. We have also created an on-the-fly version of the idea and applied it to another semantic portal Orava[6][9]. This portal provides the user with semantic search and browsing facilities similar to MuseumFinland but to a database of some 2200 video and audio clips[7] and learning object metadata (LOM)[8] related to them.

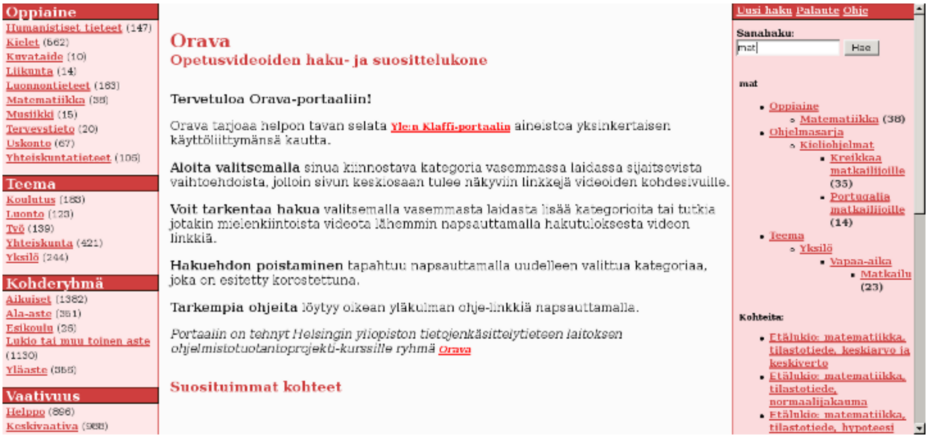
Figure 2 depicts the home page of the portal with the on-the-fly semantic autocompletion in action in the upper right corner. The user has typed in the characters 'mat', aiming perhaps at the word 'matkailu' (travel). The autocompletion function dynamically and automatically updates the category trees below as selectable links. It shows all facet categories matching the typed characters used in the multi-facet search. The facets, such as 'Oppiaine' (learning subject) and 'Teema' (theme), and their uppermost levels of subcategories are seen on the left hand side column.

Continuing by typing the letter 'k' would eliminate the category 'matematiikka' (mathematics) as no longer matching, updating the trees accordingly. Alternately, at any point the user can select a link in the dynamic facet, and the system retrieves all material related to the selected category or any of its subcategories. The presentation of the retrieved categories as trees gives the user the context necessary to make informed selections, as well as makes it possible to make a broader search by selecting some supercategory of the ones matched.

[6] http://www.museosuomi.fi/orava/

[7] The material is from the Klaffi portal (http://www.yle.fi/klaffi/) of the Finnish Broadcasting Company YLE.

[8] http://ltsc.ieee.org/wg12/



Oppiaine

- Humanistiset tieteet (147)
- Kielet (662)
- Kuvataide (10)
- Liikunta (14)
- Luonnontieteet (163)
- Matematiikka (38)
- Musiikki (15)
- Terveystieto (20)
- Uskonto (67)
- Yhteiskuntatieteet (105)

Teema

- Koulutus (183)
- Luonto (123)
- Työ (139)
- Yhteiskunta (421)
- Yksilö (244)

Kohderyhmä

- Aikuiset (1382)
- Ala-aste (351)
- Esikoulu (26)
- Lukio tai muu toinen aste (1130)
- Yläaste (355)

Vaativuus

- Helppo (896)
- Keskivaativa (988)

Orava

Opetusvideoiden haku- ja suosittelukone

Tervetuloa Orava-portaaliin!

Orava tarjoaa helpon tavan selata Yle:n Klaffi-portaalin aineistoa yksinkertaisen käyttöliittymänsä kautta.

Aloita valitsemalla sinua kiinnostava kategoria vasemmassa laidassa sijaitsevista vaihtoehdoista, jolloin sivun keskiosaan tulee näkyviin linkkejä videoiden kohdesivuille.

Voit tarkentaa hakua valitsemalla vasemmasta laidasta lisää kategorioita tai tutkia jotakin mielenkiintoista videota lähemmin napsauttamalla hakutuloksesta videon linkkiä.

Hakuehdon poistaminen tapahtuu napsauttamalla uudelleen valittua kategoriaa, joka on esitetty korostettuna.

Tarkempia ohjeita löytyy oikean yläkulman ohje-linkkiä napsauttamalla.

Portaalin on tehnyt Helsingin yliopiston tietojenkäsittelytieteen laitoksen ohjelmistotuotantoprojekti-kurssille ryhmä Orava

Suosituimmat kohteet

Sanahaku: mat

mat

- Oppiaine
  - Matematiikka (38)
- Ohjelmasarja
  - Kieliohjelmat
    - Kreikkaa matkailijoille (35)
    - Portugalia matkailijoille (14)
- Teema
  - Yksilö
    - Vapaa-aika
      - Matkailu (23)

Kohteita:

- Etälukio: matematiikka, tilastotiede, keskiarvo ja keskiverto
- Etälukio: matematiikka, tilastotiede, normaalijakauma
- Etälukio: matematiikka, tilastotiede, hypoteesi

Fig. 2. Semantic autocompletion on-the-fly in Orava

Below the dynamic autocompleted category tree, a dynamic hit list that consists of the union of all video and audio clips matching 'mat' is also shown for the direct selection of a particular item. As in MuseumFinland, autocompletion is here extended to actually searching the contents, but this time on-the-fly.

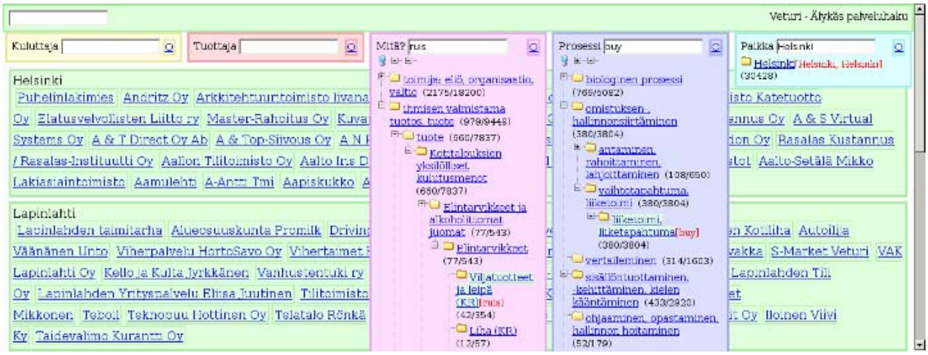
3.3 Semantic Autocompletion Facet by Facet: Case Veturi

In the semantic yellow page portal Veturi [10], created in the Intelligent Web Services (IWebS) project[9], the integration between view hierarchy based search and on-the-fly semantic autocompletion is taken even further. For this portal, on-the-fly semantic autocompletion was chosen as the central user interface element. The portal makes ample use of otherwise invisible metadata to match typed-in keywords to categories, as will be shown below.

Figure 3 depicts the search interface of the Veturi portal. The five view-facets used in the portal are Consumer ('Kuluttaja'), Producer ('Tuottaja'), Target ('Mitä?'), Process ('Prosessi'), and Location of the Service ('Paikka'). The views are located on the top horizontally, initially marked only by their name and an empty keyword field. Typing search terms in the fields immediately opens the corresponding facet to show matching categories available for selection. After such a selection, the facet closes again, showing only what was selected, while the results view below the facets dynamically updates to show relevant hits. For quick searches, a globally effective keyword search box is provided in the upper left corner of the interface. In this box its is possible to write a sequence of (possible partial) keywords, e.g. 'buy marmelade', that are completed one after another against the views.

The example search depicted in figure 3 shows the user trying to find out where he can buy rye bread in Helsinki. He has already selected Helsinki as the

[9] http://www.seco.tkk.fi/projects/iwebs/



Veturi - Älykäs palveluhaku

Kuluttaja

Tuottaja

Mitä? ruis

Prosessi buy

Paikka Helsinki

Helsinki[Helsinki, Helsinki]
(30428)
toimija: eliö, organisaatio, valtio (2176/18200)
ihmisen valmistama tuotos, tuote (979/9448)
tuote (560/7837)
Kotitalouksien yksilölliset kulutusmenot (680/7837)
Elintarvikkeet ja alkoholittomat juomat (77/543)
Elintarvikkeet (77/543)
Viljatuotteet ja leipä (KR)[ruis] (42/354)
Liha (KR) (12/57)
biologinen prosessi (769/5082)
omistuksen-, hallinnonsiirtäminen (380/3804)
antaminen, rahoittaminen, lahjoittaminen (108/650)
vaihtotapahtuma, liiketoimi (380/3804)
liiketoimi, liiketapahtuma[buy] (380/3804)
vertaileminen (314/1603)
sisällöntuottaminen, -kehittäminen, kielen kääntäminen (433/2920)
ohjaaminen, opastaminen, hallinnon hoitaminen (52/179)
Helsinki
Puhelinlaitimies
Andritz Oy
Arkkitehtuuritoimisto Iivana
Oy
Elatusvelvollisten Liitto ry
Master-Rahoitus Oy
Kuva
Systems Oy
A & T Direct Oy Ab
A & Top-Siivous Oy
A N
/ Rasalas-Instituutti Oy
Aallon Tilitoimisto Oy
Aalto Iris D
Lakiasiaintoimisto
Aamulehti
A-Antti Tmi
Aapiskukko
isto Katetuotto
nnus Oy
A & S Virtual
don Oy
Rasalas Kustannus
stot
Aalto-Setälä Mikko
Lapinlahti
Lapinlahden taimitarha
Alueosuuskunta Promilk
Driving
Väänänen Unto
Viherpalvelu HortoSavo Oy
Vihertaimet
Lapinlahti Oy
Kello ja Kulta Jyrkkänen
Vanhustentuki ry
Oy
Lapinlahden Yrityspalvelu Elisa Juutinen
Tilitoimisto
Mikkonen
Teboil
Teknobuu Hottinen Oy
Telatalo Rönkä
Ky
Taidevalimo Kurantti Oy
n Kotiliha
Autoilija
vakka
S-Market Veturi
VAK
Lapinlahden Til
et
t Oy
Iloinen Viivi

Fig. 3. Semantic autocompletion on-the-fly in Veturi

locale for the services he requires. Now, he is in the process of describing the actual service.

In the view Target view ('Mitä?'), the user has typed in the word 'rye' ('ruis'). While the annotation ontology used does not contain different grains, the concept 'grain products and bread' ('Viljatuotteet ja Leipä (KR)') contains a textual reference to rye, resulting in a category match. In this way, existing textual material can be used to augment incomplete ontologies to at least return some hits for concepts that have not yet been added into the ontology. Showing such hits in their ontological context allows for easy spotting of irrelevant hits and close misses, where for example the keyword matches a subcategory of a more appropriate one.

The search query entered in the view Process ('Prosessi') divulges another feature of semantic autocompletion: multilanguage support. Typing in the word 'buy' matches the appropriate business transaction, even though the word for 'buy' in Finnish would be 'ostaa'.

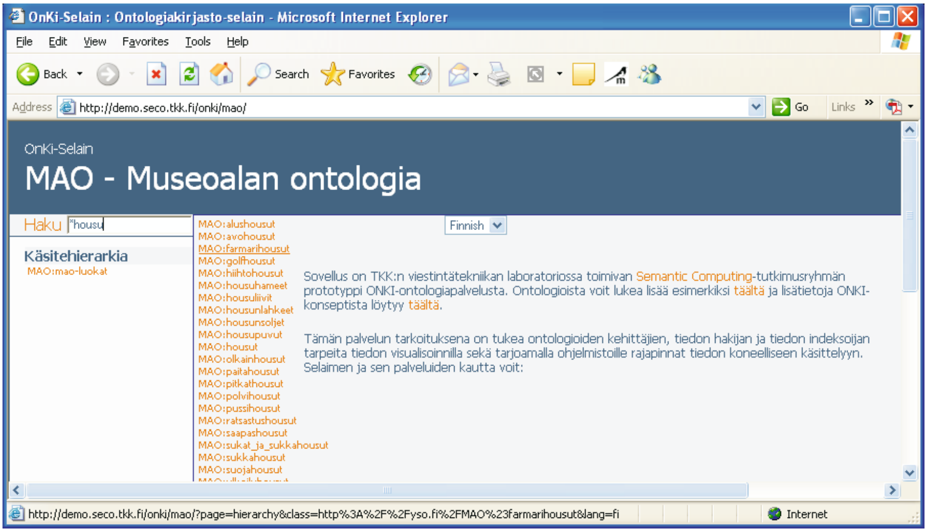
3.4 Semantic Indexing: Case ONKI Ontology Server

ONKI [11] is a part of the "Finnish National Ontologies on the Semantic We" (FinnONTO)[10] framework project. Its goal is to support the development and use of nationally shared ontologies in order to enhance semantic interoperability on the Finnish semantic web. A central part of FinnONTO research deals with providing ontology services through public web services. For a content indexer, the ONKI ontology server[11] provides a web-based browser for finding desired concepts. Semantic autocompletion has been implemented as a part of a demonstrational ONKI service.

The interface is analogous to the one in the Orava portal. In figure 4, the user has typed in the regular expression '*housu' (trouser), where '*' matches any sequence of characters, and ONKI browser has completed the input into several

[10] http://www.seco.tkk.fi/projects/finnonto/

[11] http://www.seco.tkk.fi/applications/onki/


OnKi-Selain : Ontologiakirjasto-selain - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Internet

http://demo.seco.tkk.fi/onki/mao/

OnKi-Selain

MAO - Museoalan ontologia

Haku *housu

Käsitehierarkia

MAO:mao-luokat

MAO:alushousut
MAO:avohousut
MAO:farmarihousut
MAO:golfhousut
MAO:hiihtohousut
MAO:housuhameet
MAO:housuliivit
MAO:housunlahkeet
MAO:housunsoljet
MAO:housupuvut
MAO:housut
MAO:olkainhousut
MAO:paitahousut
MAO:pitkathousut
MAO:polvihousut
MAO:pussihousut
MAO:ratsastushousut
MAO:saapashousut
MAO:sukat_ja_sukkahousut
MAO:sukkahousut
MAO:suojahousut

Finnish

Sovellus on TKK:n viestintätekniikan laboratoriossa toimivan Semantic Computing-tutkimusryhmän prototyppi ONKI-ontologiapalvelusta. Ontologioista voit lukea lisää esimerkiksi täältä ja lisätietoja ONKI-konseptista löytyy täältä.

Tämän palvelun tarkoituksena on tukea ontologioiden kehittäjien, tiedon hakijan ja tiedon indeksoijan tarpeita tiedon visualisoinnilla sekä tarjoamalla ohjelmistoille rajapinnat tiedon koneelliseen käsittelyyn. Selaimen ja sen palveluiden kautta voit:

http://demo.seco.tkk.fi/onki/mao/?page=hierarchy&class=http%3A%2F%2Fyso.fi%2FMAO%23farmarihousut&lang=fi

Fig. 4. Semantic autocompletion in the ontology server ONKI

concept categories of different types of trousers defined in the underlying cultural ontology MAO of the MuseumFinland portal. After selecting a concept by clicking on, the semantic neighborhood of the concept can be browsed further, if needed. Using ONKI, data of the selected concept such as label and the corresponding URI can read into an external application via a web service interface. ONKI can in this way be used as a service for accurate semantic indexing.

4 Implementation

The portals discussed are based on the semantic portal tool OntoViews [3], and share the same implementation of semantic autocompletion. In the implementation, the user interface component is a shallow HTML/JavaScript wrapper, whose only responsibility is to forward typed keypresses to the server. In MuseumFinland the user interface elements are static HTML, but all the newer on-the-fly implementations make use of Ajax (Asynchronous JavaScript and XML) and the XMLHttpRequest-object[12] technologies to make HTTP queries to the server in the background while viewing a page. Depending on the complexity of the user interface, the returned content is either simple HTML to be added to the page, or JavaScript code to be executed in the context of the page.

In OntoViews, all the actual keyword matching is done on the server by Ontogator [12], the view-based search engine of OntoViews. This gives the benefit of tight integration with the main multi-facet search facilities of the engine. The search is accomplished as follows:

[12] See e.g. http://en.wikipedia.org/wiki/AJAX

Firstly, the complex ontological mapping, navigation and processing associated with semantic autocompletion is accomplished as a precalculation, alongside the view projection for the multi-facet search. For each category to be projected, a set of logic rules expressed in Prolog is consulted that dictate which labels of which ontological entities are to be associated with that category. By using such rules, the ontology manipulation involved is abstracted into chunks that are quite general, as well as easy to understand, combine and implement. For example, the Veturi system includes the following rules:

```
annotation(Category,Value):- rdf(Category,'rdfs:comment',Value).

annotation(Category,Value) :-
    sumoclass(Category), rdfs_subclassof(Category,SubCategory),
    not_projected(SubCategory), annotation(SubCategory,Value).
```

The first rule states that for all classes, also their rdfs:comment should be indexed for keyword search. The second rule then states that for each class to be projected, any annotations of subclasses *not* projected will be added. In Veturi, these two rules result in adding to the quite abstract descriptions Suggested Upper Merged Ontology (SUMO) classes used, more concrete descriptions from the mid level ontology MILO that provides example subclasses for the SUMO concepts.

At runtime, the system does only very limited processing, mostly just character manipulation of the query string, such as expanding T9-type ambiguous numerical queries [1,2] to their possible extensions. Done this way, semantic autocompletion can easily be combined with other advances in predictive text autocompletion, because the ontological navigation happens completely separately from any string matching, similarly to the approach described in [13].

5 Discussion

This paper introduced the idea of semantic autocompletion as a natural extension to traditional autocompletion based on string matching. The idea is to use semantic structures for completing user text input into semantically relevant choices based on the underlying ontologies and content. Several forms of semantic autocompletion were proposed using equivalence relations, indirect semantic relations, semantic roles, and the idea extends seamlessly into semantic search. Semantic autocompletion uses not only string matching but also logical reasoning based on the underlying ontological structures. From the end-users viewpoint the matching occurs on the semantic level. The input text and completed choice labels may be quite different, but their relation to the query can still be understood and useful.

Our implementations and practical application of the idea to multi-facet search in semantic portals suggest that semantic autocompletion should be of practical value on the semantic web. Comprehensive user testing of the approach has not been done yet. However, the intuition obtained in implementing and expanding the view-based user interfaces to support semantic autocompletion point to good results. Combining keyword searching to the visualization capabilities of the facet hierarchies gives the user a quick path into the system, and gives at the same time an overview of what kind of information there is in the vocabulary. This guides the user in formulating the query in terms of appropriate concepts. Furthermore, showing hits inside the hierarchies solves the problems of homonymous query terms: the right meaning can be disambiguated by the view context.

Dealing with large and deep hierarchies is a major bottleneck of the multi-facet search paradigm. According to user tests [14], keyword search is usually preferred over multi-facet search if the user is capable of expressing her information need terms of accurate keywords. Semantic autocompletion makes it easier to the end-user to deal the wealth of categories used in facets. The value of semantic autocompletion here comes from the integration of the benefits of the keyword-based and multi-facet search paradigms.

Acknowledgements

Samppa Saarela was responsible for creating the server-side search engine used in the OntoViews framework, and Teppo Känsälä integrated autocompletion on-the-fly with Orava. Semantic autocompletion for the ONKI ontology server was implemented by Ville Komulainen.

Our work is a part the "National Semantic Web Ontology Project in Finland (FinnONTO)[13]", funded by the National Funding Agency for Technology and Innovation (Tekes) and a consortium of some 30 public organizations and companies.

References

1. Dunlop, M.D., Crossan, A.: Predictive text entry methods for mobile phones. Personal Technologies **4** (2000)
2. Hasselgren, J., Montnemery, E., Nugues, P., Svensson, M.: Hms: A predictive text entry method using bigrams. In: Proceedings of the Workshop on Language Modeling for Text Entry Methods, 10th Conference of the European Chapter of the Association of Computational Linguistics, Budapest, Hungary, Association for Computational Linguistics (2003) 43–49
3. Mäkelä, E., Hyvönen, E., Saarela, S., Viljanen, K.: Ontoviews—a tool for creating semantic web portals. In: Proceedings of the 3rd International Semantic Web Conference (ISWC 2004), Hiroshima, Japan, Springer–Verlag, Berlin (2004) 797–811
4. Pollitt, A.S.: The key role of classification and indexing in view-based searching. Technical report, University of Huddersfield, UK (1998) http://www.ifla.org/IV/ifla63/63polst.pdf
5. Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., Lee, K.P.: Finding the flow in web site search. CACM **45** (2002) 42–49

[13] http://www.seco.tkk.fi/projects/finnonto/

6. Hyvönen, E., Saarela, S., Viljanen, K.: Application of ontology-based techniques to view-based semantic search and browsing. In: The semantic web: research and applications. First European Semantic Web Symposium, ESWS 2004, Heraklion, Greece, Springer–Verlag, Berlin (2004) 92–106.
7. Hyvönen, E., Mäkelä, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., Kettula, S.: MuseumFinland—Finnish Museums on the Semantic Web. Journal of Web Semantics **3** (2005)
8. Hyvönen, E., Junnila, M., Kettula, S., Mäkelä, E., Saarela, S., Salminen, M., Syreeni, A., Valo, A., Viljanen, K.: Finnish Museums on the Semantic Web. User's perspective on MuseumFinland. In: Proceedings of Museums and the Web 2004 (MW2004), Seleted Papers, Arlington, Virginia, USA. (2004) http://www.archimuse.com/mw2004/papers/hyvonen/hyvonen.html.
9. Känsälä, T., Hyvönen, E.: A semantic view-based portal utilizing Learning Object Metadata. Paper, submitted, http://www.seco.hut.fi/publications/2006/kansala-hyvonen-2006-semantic-portal-lom.pdf (2006)
10. Mäkelä, E., Viljanen, K., Lindgren, P., Laukkanen, M., Hyvönen, E.: Semantic yellow page service discovery: The veturi portal. In: Proceedings of the 4rd International Semantic Web Conference (ISWC 2005), Poster papers, Galway, Ireland. (2005)
11. Valo, A., Hyvönen, E., Komulainen, V.: A collaborative ontology development and service framework ONKI. In: Proceedings of Int. Conf. on Dublin Core and Metadata Application (DC-2005), Madrid. (2005)
12. Mäkelä, E., Hyvönen, E., Saarela, S.: Ontogator—a semantic view-based search engine service for web applications. Paper, submitted, http://www.seco.hut.fi/publications/2006/makela-hyvonen-saarela-ontogator-2006.pdf (2006)
13. Legrand, S., Tyrväinen, P., Saarikoski, H.: Bridging the word disambiguation gap with the help of OWL and semantic web ontologies. In: Proceedings of the Workshop on Ontologies and Information Extraction, Eurolan 2003. (2003) 29–35
14. English, J., Hearst, M., Sinha, R., Swearingen, K., Lee, K.P.: Flexible search and navigation using faceted metadata. Technical report, University of Berkeley, School of Information Management and Systems (2003)

Ubiquitous Metadata Scouter – Ontology Brings Blogs Outside

Takahiro Kawamura[1], Shinichi Nagano[1], Masumi Inaba[1], Tetsuo Hasegawa[1], and Akihiko Ohsuga[1]

Research and Development Center, Toshiba Corp.

Abstract. In this paper, we introduce a service where ontology summarizes blogs to get useful in the real stores. In ubiquitous computing environment, it would be desired for users to bind their real world situation and useful information on the Internet. However, the current typical device for ubiquitous computing like a cellular phone has a small display, limited operations, and narrow-band network. Therefore, semantics use to extract only the necessary information and services is for the ubiquitous computing. Ubiquitous Metadata Scouter is for the user to scan products barcodes by cameras of cellular phones. It gets the corresponding metadata to the product from the Internet, and collect the related blogs. Then, it analyzes the contents of each blog referring ontologies, and indicates the total reputation. Also, it shows other related products which are much talked about. This paper illustrates each function of this service and our public experiment at the real consumer electronics store and book store in Tokyo since March 2006. It would provide an instant benefit as a semantics usecase in the ubiquitous computing.

1 Introduction

In ubiquitous computing environment, it would be desired for users to bind their real world situation and useful information on the Internet. The typical example of this sort of cocept a.k.a real world connection is like showing movie information if the user is heading a theater, or suggesting sightseeing points near there if the user is now traveling. In the conventional desktop computing, the information system is allowed to show lots of results at once to the user, because most of users have big displays, familiar interfaces, and broadband network, then it is possible to check and see them repeatedly. For example, most of people already got used to check 100 results emitted by Google, and click and click again to finally get the desired information.

However, typical devices for ubiquitous (currently, just mobile) computing like cellular phones have small displays, limited operations, and narrowband network (in comparison with PC). Therefore, in order to get the useful information in the Internet by a relatively easier way, it must make the operations more automatic, and indicate the really necessary information to the user. In fact, a survey on cellular phone carriers shows that the number of Internet use by cellular phones is hitting the ceiling for these years.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 752–761, 2006.
© Springer-Verlag Berlin Heidelberg 2006

So that, we believe that metadata and ontology use to extract only the necessary information and services for users based on their semantics is for the ubiquitous computing. We are trying to consolidate several domain ontologies for these years, and now we have Japanese ontologies with more than 100,000 concepts and in-house libraries to quickly search and operate those multi-bytes ontologies. In this paper, we introduce a practical usecase in which we applied these ontologies to information retrieval technique in ubiquitous computing, called Ubiquitous Metadata Scouter[1].

2 What Is Ubiquitous Metadata Scouter

Recently, bloggers who quickly are checking press releases concerning new products, then publishing reviews like comparison and utility from their own view points on their blogs have multiplied exponentially. On the other hand, the users who are considering to buy something tend to refer their blogs, then come up with their decision on purchase. Further, those users would also become the bloggers and publish their consideration. Consequently, word-of-mouth information grows rapidly.

Ubiquitous Metadata Scouter is for the user to scan products barcodes by cameras of cellular phones (at least in Japan, more than 90% of new phones have cameras). It gets the corresponding metadata to that product from the Internet, and collects the related blogs (weblogs). Then, it shows the summarized word-of-mouth information to the user in real time. For example, if the user snaps a barcode of a book, it finds a metadata including its book title, publisher, author via UPC/EAN/JAN or ISBN, and collects blogs mentioning the book reviews. Then, it analyzes the contents of each blog referring the Japanese ontologies, and indicates the total reputation (Positive / Negative determination). Also, it shows other related products which are much talked about (Hot Topic extraction). Then, it puts some of blogs which seem to be worth reading (Sort and Filtering). Fig. 1 shows an usecase of Ubiquitous Metadata Scouter. Further, fig. 2 shows an example output for a product. In the following sections, we briefly introduce the above three functions.

2.1 Positive / Negative Determination

There are several techniques called Positive / Negative determination in natural language summarization researches. A typical way is to retrieve triples $< subject, attribute, value >$ like $< car, speed, fast >$ for certain target words (subject), by checking modification relation among word classes through morphological analysis and syntactic parsing on sentences[2]. Further, there are several applied techniques like extension of variety of documents and style of output. For example, document class is changed to rating comments in an auction site[3], and the result is represented in a radar chart with axes for each attribute. Basically, we have put the following two improvements on top of these conventional researches in order to apply it for the blogs.

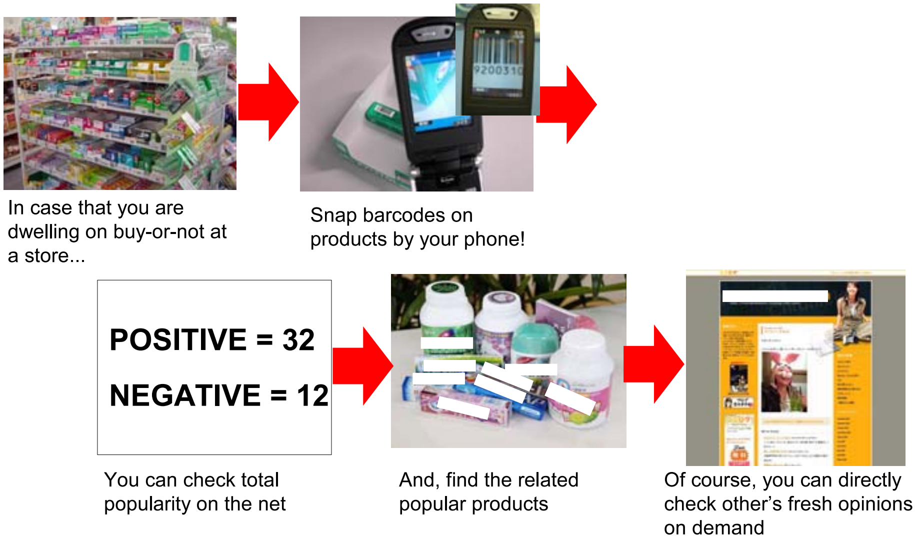

In case that you are dwelling on buy-or-not at a store...
Snap barcodes on products by your phone!
POSITIVE = 32
NEGATIVE = 12
You can check total popularity on the net
And, find the related popular products
Of course, you can directly check other's fresh opinions on demand

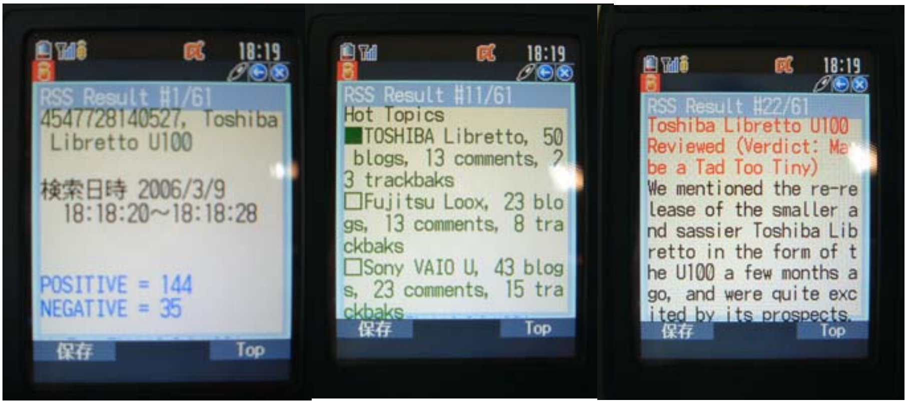
Fig. 1. Usecase of Ubiquitous Metadata Scouter

The first one is to put some weights as the importance of each blog and bias the total evaluation, considering the correlation among blog entries (each article in blogs) measured by blog metadata, RSS (RDF Site Summary) 1.0[4]. In most of previous researches, the target documents are a set of documents prepared as a corpus, and the individual situation that the user will encounter several opinions by tracking the links of the web pages is out of their scope. However, for example, an opinion in a blog which has lots of trackbacks (a function to notify to the blog author saying I have put a link to your blog entry) in favor and one with no trackback would be different on psychological impact for the user, in practical.

There are some researches using trackbacks as a web mining technique[5], but no research actively making use of implicit but useful links retrieved by RSS, such as mappings between certain blog authors and their repeated interests, and opinion flows of pros and cons on trackback contents. The blog is not a home page, nor yet-another ad. space, but a loosely-coupled community. So that, for example, an opinion of an author submitting many reviews on a product domain and one of a chance author would be different on psychological impact for the user.

Therefore, our p/n determination has limited target documents to blogs, and focused on extract and utilize the correlation of them gotten by their RSSs. Specifically, we are using the following weighting heuristics. Fig. 3 illustrates the correlation among blogs.

1. put the weight on opinions by trackbacks rather than comments. (Non anonymity, b2 and b3)
2. put the weight according to the amount of trackbacks and comments in favor from different authors. (Widely acceptance, b6)



```
RSS Result #1/61
4547728140527, Toshiba
 Libretto U100

検索日時 2006/3/9
  18:18:20~18:18:28

POSITIVE = 144
NEGATIVE = 35
保存                Top
```

```
RSS Result #11/61
Hot Topics
■TOSHIBA Libretto, 50
 blogs, 13 comments, 2
3 trackbaks
□Fujitsu Loox, 23 blo
gs, 13 comments, 8 tra
ckbaks
□Sony VAIO U, 43 blog
s, 23 comments, 15 tra
ckbaks
保存                Top
```

```
RSS Result #22/61
Toshiba Libretto U100
Reviewed (Verdict: Ma
be a Tad Too Tiny)
We mentioned the re-re
lease of the smaller a
nd sassier Toshiba Lib
retto in the form of t
he U100 a few months a
go, and were quite exc
ited by its prospects.
保存                Top
```

Summary of Word-of-Mouth (P/N determination)

Suggestion of similar products (Hot Topic extraction)

Selected blog entries (Sort & Filtering)

Fig. 2. Example result

3. put the weight on opinions of the author reviewing other related products. (Expert, to be expected some comparison from one perspective)
4. put the weight on an agreement among the flow of disagreements in responses, that is, trackbacks and comments, vice versa. (The brave, c9 in b7)
5. put the weight on opinions which are collecting responses for a long period although its time stamp is old. (Pioneer, b8)
6. put the weight on opinions which have high value produced by dividing the time of the first response to the last one by the number of responses. (High acceleration, b9)
7. put the weight on opinions of an author whose average number of responses is high. (Opinion Leader, a1)
8. decrease the weight on blogs which have no response. (No ads, b10-12)
9. decrease the weight on blogs of authors who have lots of blogs with no response (No agency, a2)
10. put the weight on frequently exchanged opinions between a few authors. (Debate, a3 and a4)
11. Finally, according to a survey report[6] 70% of authors tend to say good aspects than bad ones. So, if it is the bad comments like claims, we determine the intension of the author is higher than good comments, then put the weight.

Needless to say, fixing of the actual weight value for each heuristics greatly affects accuracy of the p/n determination. So that, we will have the evaluation on the accuracy through public experiment in the next section.

The second improvement is to take the degree of expression into consideration referring the Japanese ontology. It determines strong and weak expression of the attribute value based on is-a relation among concepts, not to count just one point for every p/n expression. Of course, it would reverse positive and negative expression according to each attribute even if it is the same value, such as $< car, mileage, high >$

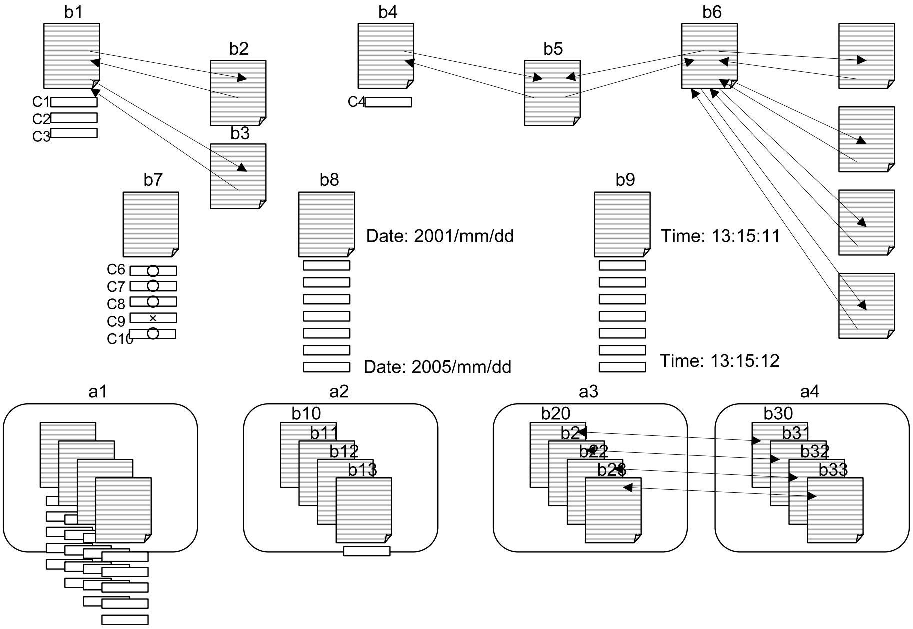


Fig. 3. Blog correlation measured by RSS metadata

and $< car, cost, high >$. Further, if the expression does not mean positive or negative directly, it might use the expression for p/n determination by referring part-of relation. Then, several other expressions for the same concept are merged by referring instance-of relation. Fig. 4 shows part of the Japanese Ontology. We apologize it's inevitably in Japanese.

Our system collects at most 100 blog entries for a certain product, then uses them for the above p/n determination. Thus, we believe that it not only correctly determines positive and negative on a document set, but also can get the p/n result similar to impression the user will have by actually browsing the blogs.

2.2 Hot Topic Extraction

Hot Topic extraction is to find other but similar products keeping attention when the user specify a certain product name. However, most of blog entries are not composed of formal sentences observing syntax, and a certain amount of blogs might be spams. So we believe that statistical work or some sort of learning techniques on frequency of specific words collected from all the blogs would fail to suggest really HOT product information.

Then, this paper made use of trackbacks as well as the above p/n determination. There are lots of cases that the users mention their own opinions in their blogs in terms of content of the others' blogs by putting trackbacks to them. This correlation among blog entries is called blog thread, where a certain topic is intensively

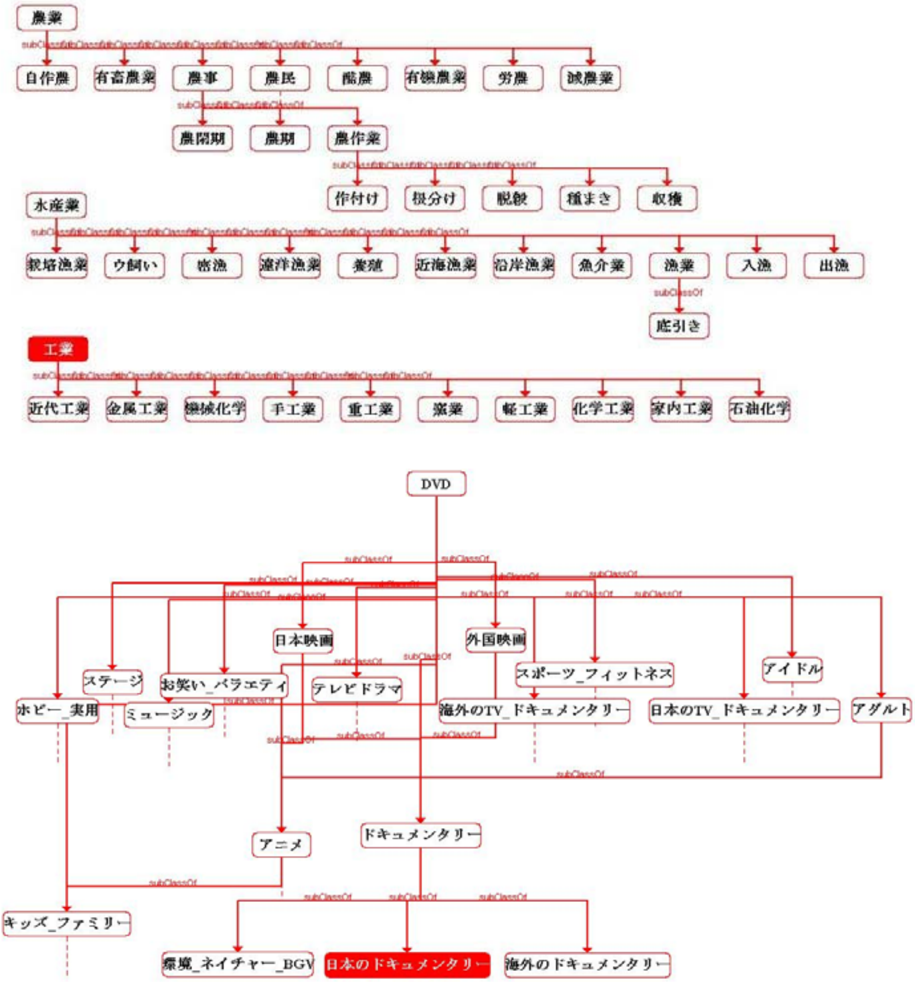


Fig. 4. Japanese Ontology (above) and Product Ontology (below)

discussed each other. Even for the third parties, they would feel more credit on opinions talked in a series of blogs which compose a blog thread in terms of a certain topic, than one in a single entry without any trackback. Therefore, to find really hot topic the approach based on the blog relationship via trackbacks seems effective.

In our system, the Hot Topic extraction consists of blog crawling and hot topic analysis. The blog crawling firstly inputs a list of product names to a blog search engine like Google Blog Search, and collects the blogs related to the products, then extracts several blog threads based on links of trackbacks on blog entries. Note that lists of products are stored in our Product Ontology as instances (individuals). We illustrated part of the Product Ontology in fig. 4. The hot topic analysis analyzes the blog threads, and calculates degree of relationship and degree of popularity on each instance. Degree of hot topic is mathematical product of both. The degree of relationship and popularity is determined as the following heuristics.

1. In a blog thread, a product talked in the first entry has a high degree of popularity.
2. In a blog thread, a product mentioned in an entry which received lots of trackbacks and comments has a high degree of popularity.
3. In a blog thread, different products in other entries than the first product in the first entry have high degrees of relationship. In addition to this, they also have high degrees of popularity according to their frequency.
4. In several blog threads, if different users comment on the same product, the product has a high degree of popularity.

As we mentioned in the previous section for p/n determination, it greatly affects on accuracy on the Hot Topic extraction how much we put the degrees based on the above heuristics. To verify it, we are now doing public experiment on the real stores in Tokyo mentioned in the next section. Then we will have evaluation on the accuracy of the Hot Topic extraction.

2.3 Sort and Filtering

The user may also want to read actual blog entries, so we show on the cellular phone MAX 20k of blog bodies which are selected to be worth reading based on the above blog correlation. For example, those are entries which received lots of trackbacks, or entries include obvious positive or negative opinions with a certain amount of sentences. On the other hand, we eliminate ads. and spams.

3 Public Experiment

Recently, we have finished development of the evaluation version. Then since March 2006, we have public experiment of this service at the real stores for consumer electronics and books in and near Tokyo. In the experiment, we are using the Product Ontology with 400,000 concepts and Product Metadata which binds barcode to product information such as title, author, manufacturer, publisher, date, etc. with 1,400,000 items, in addition to the Japanese Ontology with 100,000 concepts.

Firstly, test subjects in the experiment bring cellular phones installed our client application for Ubiquitous Metadata Scouter to the stores, then actually take some interested products on hand, snap the barcodes and check the displayed results. Next, the subjects fill out questionnaires, which includes a question asking if the results of p/n determination and Hot Topic extraction have fitted their impression after reading the actual (correlated) blog entries. Thus, we are collecting the evaluation of intuitive accuracy and deliberate accuracy, as well as performance measure like turn around time to display the result.

Currently, it takes approx. 30-40 (sec) to collect MAX 100 blogs, and do the p/n determination and Hot Topic extraction, then return and display the results with MAX 20k blog bodies on the phone. However, we should note that our server machine is just a desktop PC with Pentium 4, 3.2 GHz, Memory 1GB. Further, the initial result shows the accuracy of p/n determination is about 80% by the questionnaires, but the detailed result is not shown yet. We are now measuring the users' impression for the correlated blogs, so comparing the impression with naive positive and negative values for not-correlated blog documents, we can fix the appropriate weights for the heuristics. After that, we will show the result of the accuracy of our p/n determination soon.

The architecture of Ubiquitous Metadata Scouter is shown in fig. 5.

Snapshots of this experiment are shown in fig. 6.

Note that in order to put this system into practical use, we definitely need cooperation with real stores. We believe the followings would be the advantages for the stores.

- It could give some supportive information which prompts consumers hesitating purchase to buy a product. In fact, 70% of blog entries regarding product reviews is positive comment according to a survey[6].
- Good reputation on the net can be directly connected to sales on the real stores. On the other hand, if the stores use this system, they can have some actions in advance like stock reduction for products of bad reputation on the net.
- In terms of the Hot Topic extraction, if the stores do not have the suggested products at that time, it can be guided to order thereon.
- There is a possibility to produce new hit products from the blogs.

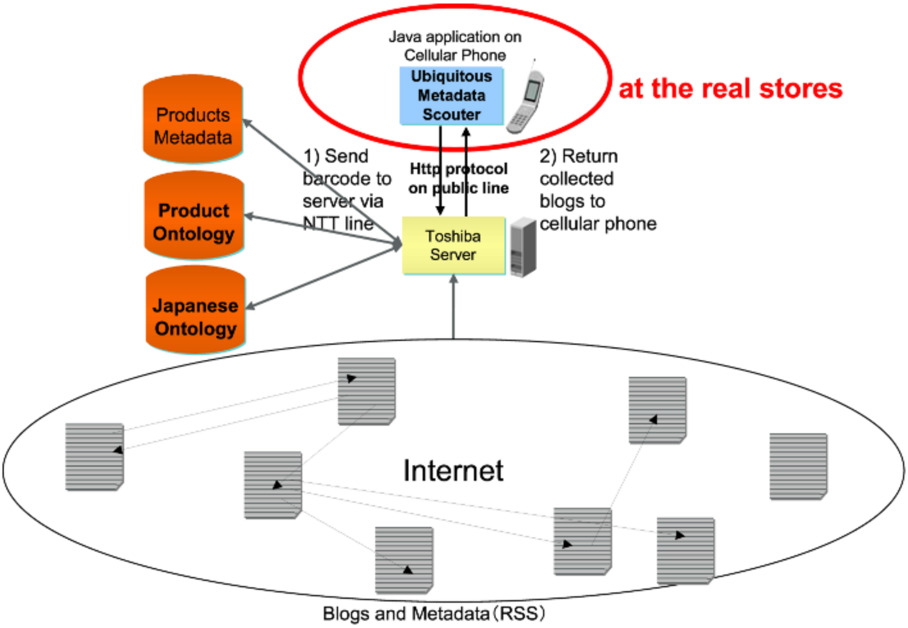

Java application on Cellular Phone
Ubiquitous Metadata Scouter
at the real stores
Products Metadata
Product Ontology
Japanese Ontology
1) Send barcode to server via NTT line
Http protocol on public line
2) Return collected blogs to cellular phone
Toshiba Server
Internet
Blogs and Metadata (RSS)

Fig. 5. System architecture

Fig. 6. Public Experiment

Further note that this service does not compare prices at each store, nor the reputation of stores. Thus, it would not result in unprofitable information for the real stores.

4 Related Work

In another project of our team called Ubiquitous Service Finder[7], we proposed a platform to directly coordinate services and information in the net and home/office appliances on the user's palm top. It was under an assumption that IC tags, DLNA or ECHONET compliant home appliances, and Web Services in the Internet will prevail soon.

In this paper, on the other hand, we proposed a service to retrieve word-of-mouth information by collecting blogs from barcodes when the user gets up a cellular phone to products. It is under the current observation that barcodes instead of IC tags, and blogs instead of semantic services have already prevailed. So that it would provide more instant benefit as a semantic use in ubiquitous computing.

One of other services which have similar configuration is Amazon Scan Search[8]. When the user scans a barcode on a book by the cellular phone, it shows the corresponding page in Amazon.com if it's there.

Also, Microsoft is under experiment of AURA (Advanced User Resource Annotation System)[9] in US. In this system, PDAs attached with CF-type barcode readers scan the barcodes on products, and search the related information by Google or eBay. Further, it has a web site where the users can freely annotate on those products.

Furthermore, Yahoo! Japan is operating a bbs site[10] to find some product by word-of-mouth information, and a map publisher ZENRIN[11] announced an internet map service to locate POIs (point of interest) like restaurants coupled with blogs' reputations.

5 Conclusion and Future Work

In this paper, we introduced Ubiquitous Metadata Scouter where ontology summarizes blogs to get useful in the real stores. As mentioned in the previous section, our first priority is now evaluation and improvement on performance like responses for multiple access at a time, and on accuracy of precision ratio based on the results of the public experiment.

As well as the popularization of blogs, annotation to real objects by IC tags and to digital data promoted by HDD recorders or iPod have been accelerated. Also, Web Services in the Internet is growing constantly, and it is expected that coordination among networked appliances in the home will become a big movement near future. In this circumstance, we hope we will provide a value-added ubiquitous solution with semantics.

References

1. USA Today, http://www.usatoday.com/tech/news/techinnovations/2006-02-15-bar-code-phones_x.htm?POE=TECISVA
2. N. Kobayashi, K. Inui, Y. Matsumoto, K. Tateishi, T. Fukushima, Collecting Evaluative Expressions for Opinion Extraction, First International Joint Conference (IJC-NLP 2004), LNAI 3248, 2005.
3. Y. Kusumura, Y. Hijikata, S. Nishida, NTM-Agent:Text Mining Agent for Net Auction, IEICE Transactions of Information and Systems, Vol.E87-D, No.6, pp.1386-1396, 2004.
4. D. Brickley, et al., RDF Site Summary (RSS) 1.0 http://purl.org/rss/1.0/spec, 2000.
5. M. Kimura, K. Saito, K. Kazama, S. Sato, Detecting Search Engine Spam from a Trackback Network in Blogspace, Proceedings of 9th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES2005), 2005.
6. Web Advertising Bureau, http://www.wab.ne.jp/english/index.html
7. T. Kawamura, K. Ueno, S. Nagano, T. Hasegawa, A. Ohsuga, Ubiquitous Service Finder - Discovery of Services semantically derived from metadata in Ubiquitous Computing, Proceedings of 4th International Semantic Web Conference (ISWC 2005), 2005.
8. http://www.amazon.co.jp/exec/obidos/tg/feature/-/546374/249-9948758-5416367
9. A.J. Bernheim Brush, Tammara Combs Turner, Marc A. Smith, Neeti Gupta, Scanning Objects in the Wild: Assessing an Object Triggered Information System, Proceedings of 7th International Conference on Ubiquitous Computing (UbiComp 2005), 2005.
10. http://contents.shopping.yahoo.co.jp/info/kaicom/
11. http://www.zenrin.co.jp/english/

Networked Interactive Photo Annotation and Reminiscence Content Delivery

Noriaki Kuwahara[1], Kiyoshi Yasuda[1,2], Shinji Abe[1], and Kazuhiro Kuwabara[3]

[1] ATR Intelligent Robotics and Communication Laboratories
Keihanna Science City, Kyoto, Japan
{kuwahara, sabe}@atr.jp
http://www.irc.atr.jp/index.html
[2] Chiba Rosai Hospital
Tatsumidai-Higashi, Ichihara City, Chiba, Japan
fwkk5911@mb.infoweb.ne.jp
[3] Ritsumeikan University
Noji Higashi, Kusatsu City, Shiga, Japan
kuwabara@is.ritsumei.ac.jp

Abstract. This paper proposes a distributed environment for dementia care that consists of interactive photo annotation and reminiscence content delivery over the Internet by using Semantic Web technologies. We first propose a *Networked Interactive Photo Annotation* service that supports Internet-based collaborative photo annotation among a remote video author and a dementia sufferer and his or her family. This system is built on top of an authoring tool we have developed to assist in reminiscence video production by making use of photo annotation. Combined with an IP video phone, the proposed system is intended to promote conversation between the video author and the dementia sufferer as well as to annotate the shared photo. Next, we present a *Networked Interactive Reminiscence Content Delivery* service that enables a remote dialog partner to initiate communication with the users via an IP video phone in order to deliver reminiscence contents to their display and to share these contents with them.

1 Introduction

A variety of behavioral difficulties for people with dementia, such as wandering, agitation, hallucinations, and incontinence, is placing a great burden on caregivers, who are often family members. Reminiscence videos created from old photo albums of people with dementia is a promising way of maintaining their mental stability in order to alleviate such difficulties [1]. We have already proposed an authoring tool to assist in video production by using photo annotation and have shown that the reminiscence videos generated by our tool have the same effects as videos produced by human experts [2, 3].

Although our tool can reduce the rendering costs of videos by using annotation data, according to interviews with a dementia sufferer and his/her family, the cost of annotation needed to acquire an episode for each photo cannot be ignored. To reduce

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 762 – 767, 2006.
© Springer-Verlag Berlin Heidelberg 2006

this cost, we propose a service called *Networked Interactive Photo Annotation*, which uses the Internet to support a remote video author in collaborating with a dementia sufferer and his/her family in the photo annotation process. This service is the result of combining interactive photo sharing and an IP video phone [4]. In this service, photos and videos produced by our tool are stored in the reminiscence content database on the Internet, and the contents can be downloaded and played on the Web browser component of the equipment.

Although an IP video phone is designed to be easy to use, our assumed users are elderly, and operating a Web browser is still a major barrier for them. To solve this problem, we also propose a service called *Networked Interactive Reminiscence Content Delivery*. In this service, remote dialog partners support the dementia sufferers and their families by operating the Web browser remotely. In this paper, we first illustrate a scenario for each proposed service. Second, we clarify the challenges associated with implementing these services and present our approach to overcoming them by using Semantic Web Technology. Finally, we state our conclusions and mention our work in progress.

2 Proposed Service Scenarios

2.1 Networked Interactive Photo Annotation Service

Reminiscence video is a slideshow video produced from old photos of people with dementia. Our proposed tool adds a visual effect to each photo (the so-called "Ken Burns effect" [5]), narrations, and BGM (background music) to make the video more attractive.

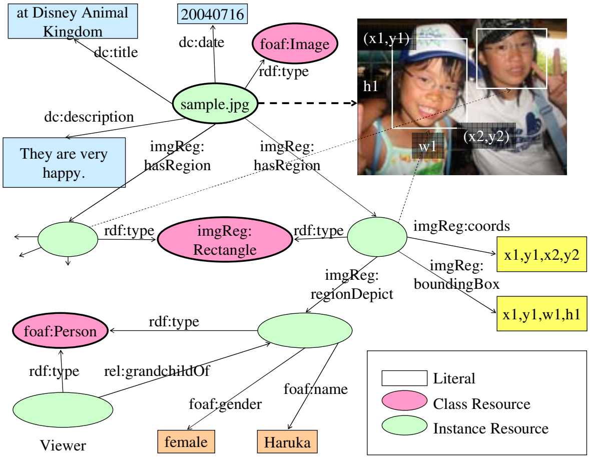


Fig. 1. Illustrative example of photo annotation data

The importance of such effects has already been experimentally demonstrated [6]. We focus on the people in the photo in adding annotation. The Dublin Core is used to describe various properties of the photo itself. FOAF is used to describe people in the

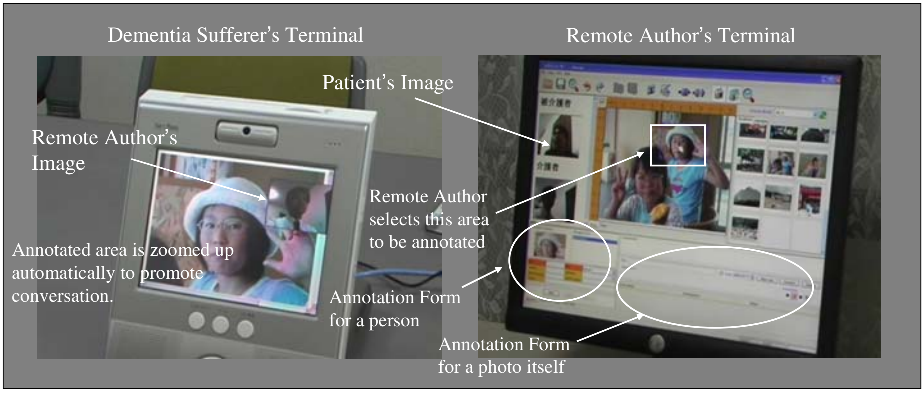

Dementia Sufferer's Terminal
Remote Author's Terminal
Patient's Image
Remote Author's Image
Remote Author selects this area to be annotated
Annotated area is zoomed up automatically to promote conversation.
Annotation Form for a person
Annotation Form for a photo itself

Fig. 2. GUI of Networked Interactive Photo Annotation Service (Photo Sharing)

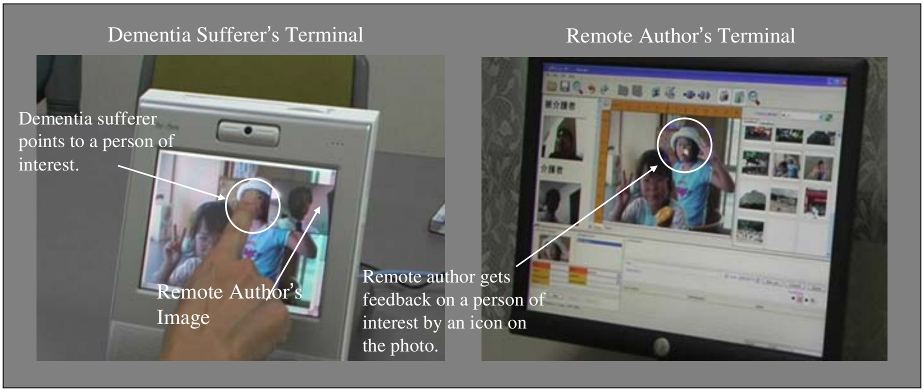

Dementia Sufferer's Terminal
Remote Author's Terminal
Dementia sufferer points to a person of interest.
Remote Author's Image
Remote author gets feedback on a person of interest by an icon on the photo.

Fig. 3. GUI of Networked Interactive Photo Annotation Service (Intuitive Interface)

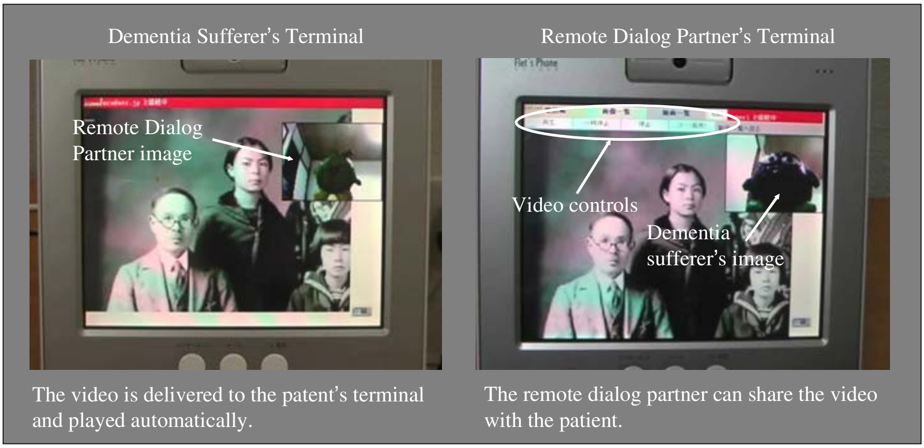
photo. Image Regions are used to store each photo's region data corresponding to a person in the photo. Figure 1 shows the illustrative example of the photo annotation.

Figure 2 shows the GUI of the *Networked Interactive Photo Annotation* service. Old photos of the dementia sufferer are stored in the reminiscence content database beforehand. The remote video author calls the dementia sufferer and his/her family over the IP video phone and selects the photo to be shared with them. Then the selected photo is automatically displayed on the dementia sufferer's terminal.

The remote video author can also pan across and zoom up on each person in the photo to help those with impaired eyesight focus their attention on each person, which is also shown in Fig. 2. Furthermore, our service provides dementia sufferers with an intuitive interface to let the remote video author know with whom the sufferer is well acquainted from among people in the photo by touching the display (Fig. 3). The touched position is transmitted to the video author's side and shown on the author's display by using the icon as also presented in Fig. 3.

2.2 Networked Interactive Reminiscence Content Delivery Service

In order to help a dementia sufferer and his/her family members to operate the Web browser of their terminal when they use reminiscence videos, the *Networked Interactive Reminiscence Content Delivery* service calls the remote dialog partner by IP video phone and requests remote operation for them. With the remote dialog partner's support, family members are able to use the video with a dementia sufferer; otherwise, the remote dialog partner can take care of the dementia sufferer by sharing such content while family members take a short respite as shown in Fig. 4.


Dementia Sufferer's Terminal
Remote Dialog Partner's Terminal
Remote Dialog Partner image
Video controls
Dementia sufferer's image

The video is delivered to the patent's terminal and played automatically.

The remote dialog partner can share the video with the patient.

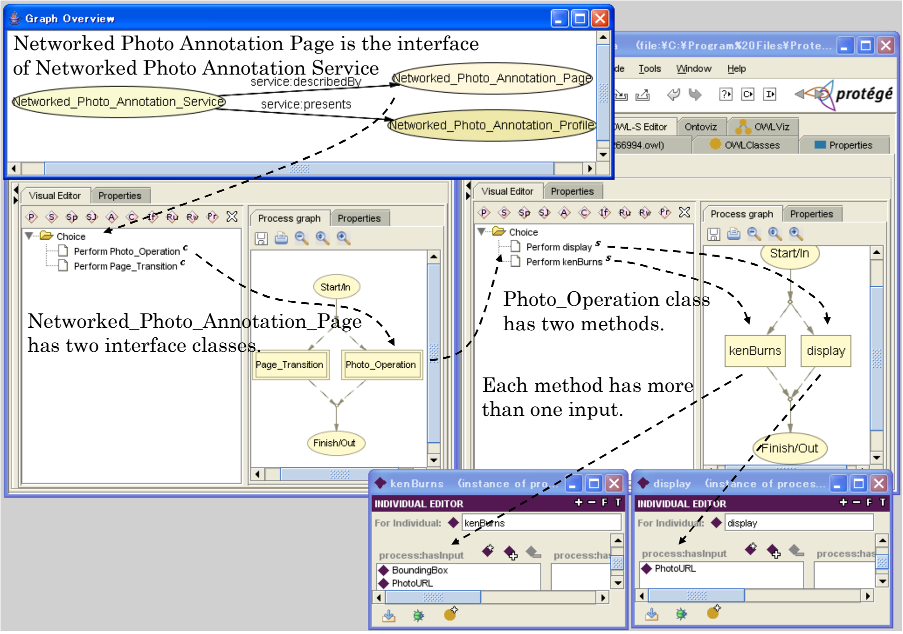
Fig. 4. GUI example of Networked Interactive Reminiscence Content Delivery Service

On remote dialog partner's display, video control menu is displayed and the remote dialog partner can select, play and stop the video on dementia sufferer's terminal by using these controls in synchronization with the video on his/her terminal so that they can watch the same scene of the video and can talk with it as if they were sitting beside each other.

3 Technical Challenges for Implementing Services

Recent IP video phones available commercially often include a Web browser function. Therefore, the user interface of our proposed services has been implemented as Web contents to be remotely operated by the dialog partner. Then, we prepared an asynchronous messaging mechanism between web contents in order to make it possible for the dialog partner to remotely operate the dementia sufferer's Web contents [7].

We considered this setup a form of coordination between remote Web contents featuring interfaces for remote operations. Consequently, to describe the remote Web content's interface, we employed the set of class definitions in OWL-S [8] through the OWL-S editor [9] (Fig. 5). Remote interface classes are represented by using CompositeProcess, and methods of each class are mapped to SimpleProcess. We prepared JavaScript libraries corresponding to these classes, which enable Web contents to be remotely operated.



Networked Photo Annotation Page is the interface of Networked Photo Annotation Service

Networked_Photo_Annotation_Page has two interface classes.

Photo_Operation class has two methods.

Each method has more than one input.

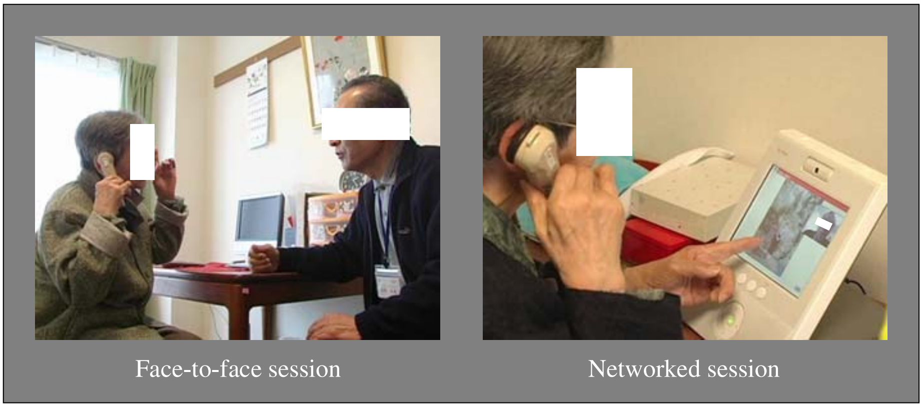
Fig. 5. OWL-S descriptions for remote Web content

4 Conclusions and Work in Progress

Our proposed services are intended for use in dementia care over the Internet. They are implemented as a combination of an IP video phone and remote reminiscence content sharing. We have come up with a messaging mechanism between Web contents in different home networks. Moreover, we have introduced a scheme for describing remote Web content interfaces by using OWL-S. Our scheme will allow service providers to develop remote services for people with dementia rapidly. This is because our approach not only improves the reusability of remote Web contents but also simplifies the process of modifying existing Web contents for remote use. Therefore, the contents originally designed for a dementia sufferer with the support of an on-site caregiver can also be later used with the support of a remote dialog partner.

We conducted an experiment under realistic conditions of networked photo annotation and reminiscence content delivery in collaboration with NTT (telecom company), Best Life Inc. (senior care home), and Association of Whole Family Care (volunteer registry of Active Listening for elderly people). Figure 6 shows experimental scenes. This subject has weak hearing and she used the bone conduction receiver in both face-to-face and networked sessions.

The experiment started in the middle of April 2006 and just finished in the middle of June 2006. We are now analyzing the data obtained through this experiment and examining whether remote dialog partners are able to collect episodes on photos properly from dementia sufferers and their family members by using our proposed services.



Face-to-face session

Networked session

Fig. 6. Experimental scenes in a senior care home

Acknowledgements

We would like to thank NTT, Best Life Inc., Association of Whole Family Care, and the participants of our experiment. We also would like to thank Mr. Koji Saito, Mr. Takeshi Ochi, and Mr. Kunihide Ikeda for the design and implementation of the services. This research was supported by the National Institute of Information and Communications Technology.

References

1. Yasuda, K. et al.: Reminiscence Video for Higher Brain Dysfunctions, Proceedings of General Conference of Japan Society for Higher Brain Dysfunctions, (2004) (in Japanese).
2. Kuwahara, N., Kuwabara, K., Tetsutani, N., and Yasuda, K.: *Reminiscence Video - Helping At-Home Caregivers of People with Dementia*, Home-Oriented Informatics and Telematics, Proceedings of the IFIP WG 9.3 HOIT 2005 Conference (2005) 145-154.
3. Kuwahara, N., Kuwabara, K., Abe, S., Yasuda, K., and Tetsutani, N.: *Semantic Synchronization: Producing Effective Reminiscence Videos*, 4th International Semantic Web Conference (ISWC2005) Demo Papers (2005).
4. http://www.fletsphone.com/index_f.html (in Japanese)
5. http://en.wikipedia.org/wiki/Ken_Burns
6. Kuwahara, N., Kuwabara, K., Abe, S., Tetsutani, N., and Yasuda, K.: *The Evaluation of Audio-Visual Effects added to a Reminiscence Video for dementia sufferers*, Proceedings of SI2005 (2005) (in Japanese).
7. Kuwahara, N., Kuwabara, K., and Abe, S.: *Networked Reminiscence Content Authoring and Delivery for Elderly People with Dementia*, Proceedings of International Workshop on Cognitive Prostheses and Assisted Communication (2006)
8. http://www.daml.org/services/owl-s/
9. http://owlseditor.semwebcentral.org/

Task-Oriented Mobile Service Recommendation Enhanced by a Situational Reasoning Engine

Takefumi Naganuma[1], Marko Luther[2], Matthias Wagner[2], Atsuki Tomioka[1],
Kunihiro Fujii[1], Yusuke Fukazawa[1], and Shoji Kurakake[1]

[1] Network Laboratories, NTT DoCoMo Inc. 3-5 Hikari-no-oka, Yokosuka-shi, Kanagawa,
239-8536 Japan
{naganuma, tomiokaa, fujiiku, fukazawayuu, kurakake}
@nttdocomo.co.jp
[2] Future Networking Lab, DoCoMo Communications Laboratories Europe GmbH
Landsbergerstr. 308-312, 80687 Munich, Germany
{luther, wagner}@docomolab-euro.com

Abstract. In this paper, we propose a system that recommends appropriate mobile services from the viewpoint of the user's task which fits with user's situation in the real world. Key components are a situation provider that reason on user situation based on context gathered from multiple sources, and a task knowledge base which stores semantic task descriptions of what actions the mobile user is likely to perform in daily life. We present the architecture of the proposed system; the situational reasoning engine which makes use of context ontologies represented using OWL, and the task knowledge base which stores OWL-S-based descriptions of the user's tasks in the real world. Finally, we describe a prototypical implementation and some realized use cases.

1 Introduction

The mobile Internet market is making rapid progress, especially in the field of mobile telephony. NTT DoCoMo is providing mobile Internet services to over 45 million mobile phone subscribers. Currently, widely diverse contents such as entertainment services (ring-tone download, games, etc), transaction services (money transfer, airline reservation, etc) and information services (weather forecast, local information, etc) are being offered through more than 98,000 mobile Internet sites [1].

We have developed a task-knowledge-based service retrieval system for the non-expert mobile user that makes it easy to retrieve services appropriate for tackling the user's problems in the real world [2]. Here, the term "task" refers to "what the user want to do" in the real world as an expression of the user's problem, and the system features a task knowledge base that contains knowledge about which services will solve the problems that the user faces in daily life. The system allows the mobile user to find services while focusing on actual user tasks in the real world. Effective for service retrieval, the system behaves passive in requiring a user's initial input to trigger the problem solving process.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 768 – 774, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In this paper, we propose a task-oriented service recommendation system based on the user's situation without require initial user input that uses Semantic Web technology. Here "situation" denotes a high-level description of a user's situation derived from applying inference mechanisms to a set of context pieces gathered from multiple context sources. The proposed system proceeds in three steps (Fig. 1). First, the user's situation (e.g. Business situation) is determined through classification-based reasoning using the underlying situation- and context ontologies. Next, possible user tasks (e.g. go shopping) that suit the situation can be selected from task knowledge base. The task knowledge base stores a lot of tasks collected from the real world and the relations between tasks. Finally, service candidates to achieve the user's task can be selected from the service knowledge base. The service knowledge base stores descriptions that associate services with tasks stored in the task knowledge base.

We will discuss the situational reasoning using OWL [3] based multiple ontologies in Section 2, and then introduce a task-knowledge modelling method and semantic description of task knowledge with OWL-S [4] in Section 3. Details are provided of the design of a prototype system including user interface on actual mobile handsets with some usage scenario in Section 4, and Section 5 draws our conclusions.

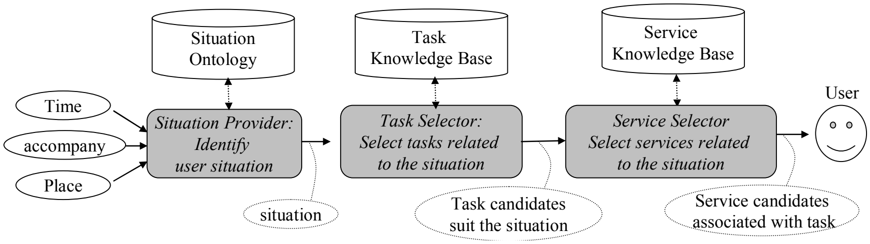

Situation Ontology
Task Knowledge Base
Service Knowledge Base
User
Time
accompany
Place
Situation Provider: Identify user situation
Task Selector: Select tasks related to the situation
Service Selector Select services related to the situation
situation
Task candidates suit the situation
Service candidates associated with task

Fig. 1. Process of task-oriented service recommendation with situational reasoning

2 Situational Reasoning with Multiple Ontologies

A logically-well-founded ontologies not only offers ways for describing a domain of interest, but also allows reasoning about the represented information [5]. We use following three context ontologies for describing actual context entities.

- *Time*: the time ontology is an adoption of the time-entry ontology of OWL-S provides the specification of time points and time intervals as well as the qualitative relations among time intervals such as "during" or "start". Furthermore, the time ontology defines abstract time concepts such as "Meal_time" or "Office_hour".
- *Space*: the space ontology provides the specification of key location and basic spatial relationships. Example concepts are "Public_place", "Private_place" on an abstract level, and "Station", "Museum" on a concrete level.
- *Agents*: the agent ontology specifies the user and related people in key roles including for instance, "Myself", "My_colleague", or the relationships such as "supervisor", "friend" or "relative".

An actual context entity is described using concepts and relations defined in the appropriate ontology [6] [7] [8]. Fig.2 shows an example of social relation description using the agent ontology. In this example, *Dawson Campbell* is related to *Madeleine Campbell* with *wife* relation and *Madeleine Campbell* is related to *Mark Buchanan* with *father* relation. In this case, *Dawson Campbell* has no direct relation with *Mark Buchanan* but we can know that *Mark Buchanan* is *Dawson Campbell*'s *father-in-law* by using ontology-based reasoning as follows;

□ The relation *father* is a sub-property of *parent*. (using "rdfs:subPropertyOf")
□ The relation *parent* is an inverse property of *child*.(using "owl:inverseOf")
□ The relation *wife* is a sub-property of *spouse*. (using "rdfs:subPropertyOf")
□ The class *(My) Parent_in_law* is the class which has a relation *child* with the class *(My) Spouse*.

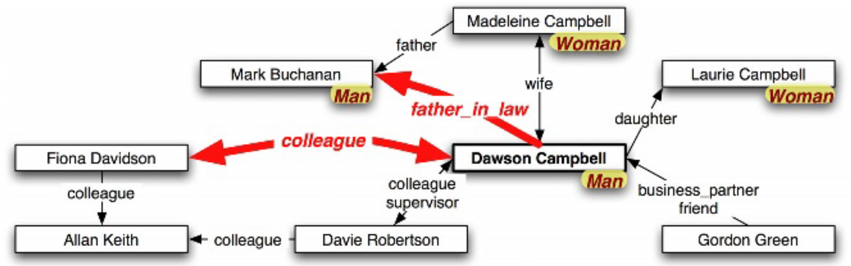


Fig. 2. Social relation description based on agent ontology

We have defined a situation ontology that allows multiple context sources to be integrated for expressing complex user's contexts. Fig.3 shows a part of this ontology. The top level concept is "Situation", and the three concepts, "Private", "Business" and "Meeting" are direct sub classes of this top level concept. The situation ontology holds logical descriptions that allow us to define concepts by using the concepts and relations defined in the multiple context ontologies. For instance, the concept "Private" is defined the user's situation such as "user is in a private place" or "user is in a public place at leisure time". Furthermore, one definition of the concept "Private_meeting" involves the intersection concept (using "owl: intersectionOf") of "Private" and "Meeting", and has a restriction (using "owl: Restriction") that "accompanies" is limited to relatives or friends.

Situational reasoning [9] is conducted by the situation ontology together with the defined context ontologies. A concrete example of such situational reasoning is the following: user A is at the station with his wife B on Sunday morning. First, each context information such as location ("station"), time ("Sunday morning"), accompanies (his wife "B") is reasoned by using context ontologies. The situation of user A is discerned as "Family_meeting" since the location is "Public_place" (because "station" is a sub concept of "Public_place"), the time is "Leisure_time" (because "Sunday morning" is classified as "Leisure_time"), and he is accompanied by just "Relative" (because his wife "B" has a relation with user A using "wife" relation, and "wife" is a sub property of "relative").

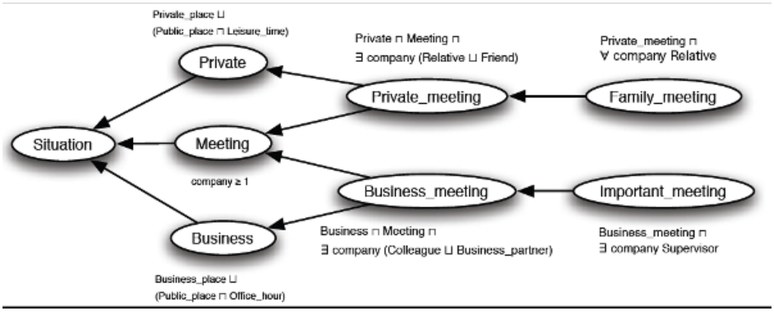


Fig. 3. A part of situation ontology

3 Task Knowledge Base

The task knowledge base is a knowledge base that stores semantic descriptions of user's tasks in the real world including abstract tasks and concrete tasks, and the relations between them. We extracted task knowledge that depends on some specific place such as an amusement park or a department store. The category of real-world places can be borrowed from commercial services such as a map service or car navigation service. We have constructed domain specific task knowledge for 30 domains so far. Fig.4 shows an example of the task knowledge entries for the department store domain.

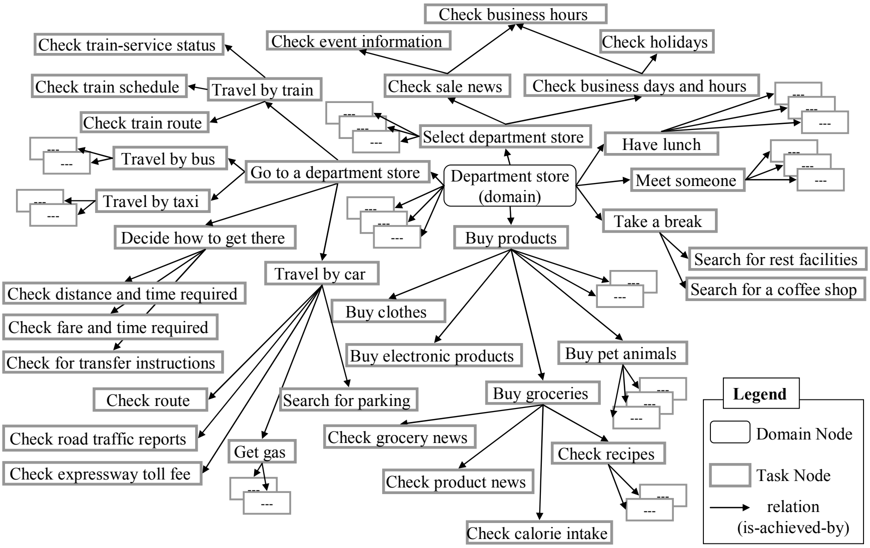


Fig. 4. An example of task knowledge base entries

We designed a description framework of task knowledge using OWL-S. OWL-S is an OWL based Web service ontology for describing the properties and capabilities of Web services. OWL-S also includes a process ontology for describing generic processes. We describe the task-knowledge structure by using the Process model and the control constructs defined in OWL-S. The context information that indicates the applicable condition of the task node is described by using the Service Profile. For instance, when the user's context is "midnight", the task of "travel by train" should not be associated with the user if there are no train services at midnight. The following is an example of a Service Profile of OWL-S which expresses the context condition using the taxonomy of situation ontology. In this code, the task ("Go Shopping") is effective for the situation of "Private_meeting".

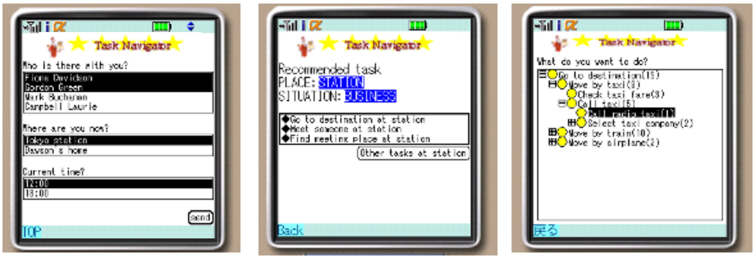
```
<profile:Profile>
<profile:serviceName>Go Shopping</profile:serviceName>
<profile:serviceParameter>
<profile:ServiceParameter>
<profile:serviceParameterName>situation</profile:serviceParameterName>
<profile:sParameter>Private_meeting </profile:sParameter>
</profile:ServiceParameter>
</profile:serviceParameter>
<profile:has_process rdf:resource="#TaskModel00000920"/>
</profile:Profile>
```

4 Implementation

We designed and implemented the proposed system that consists of server module and client module called Task Navigator running on commercial mobile phones. We show two usage scenarios and describe the user interface of the prototype system.

Scenario1: Meeting business partner at Tokyo Station
Two travelers, Dawson Campbell and his boss Fiona Davidson, arrive on a Friday morning at the main railway station of Tokyo where Gordon Green, a project partner, is waiting for them. The group are looking for a quick transfer to the airport to leave for the meeting location.

First we set 3 bits of context information, "company", "location", "time", in the context emulation window (Fig. 5 left). Next, all bits of context information are sent to the Situation Provider, and the Situation Provider then determines the user's situation. The Situation Provider's understanding is that "Fiona Davidson" and "Gordon Green" are "business partners", and "Tokyo station" is a "Public place" because "Tokyo Station" is a sub-concept of "Railway_station" which is a sub-concept of "Public_place", and the time "12:00" is a "business_hour"; according to situational reasoning, the user's situation is determined to be "Business_meeting". The value of the user's situation corresponds to Task Selector as a response, and then Task Selector selects only those tasks associated with "Business_meeting" from task knowledge base and sends those to Task Navigator. Task Navigator shows the list of possible tasks on the display (Fig. 5 center) , "Go to destination at station", "Meet someone at station", "Find meeting place at station", and the user select the most appropriate one from the list and operate the task structure related to the selected task (Fig. 5 right).



Task Navigator
Who is there with you?
Fiona Davidson
Gordon Green
Mark Buchanan
Campbell Laurie
Where are you now?
Tokyo station
Dawson's home
Current time?
12:00
18:00
send
TOP

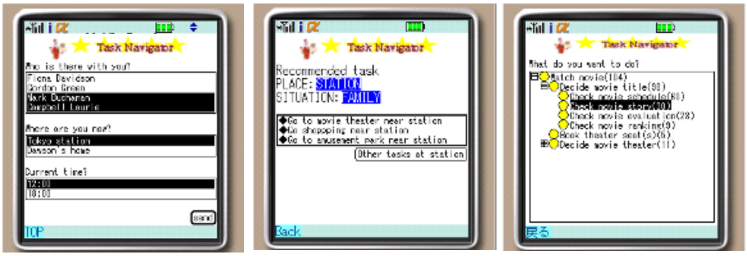
Task Navigator
Recommended task
PLACE: STATION
SITUATION: BUSINESS
◆Go to destination at station
◆Meet someone at station
◆Find meeting place at station
Other tasks at station
Back

Task Navigator
What do you want to do?
⊟Go to destination(15)
⊟Move by taxi(8)
Check taxi fare(3)
⊟Call taxi(5)
Call radio taxi(1)
⊞Select taxi company(2)
⊞Move by train(10)
⊞Move by airplane(2)
戻る

Fig. 5. Flow of service recommendation for meeting business partner at station

Scenario2: Meeting family and relative at Tokyo station
Dawson Campbell arrives around noon at the main station of Tokyo where his daughter Dawson Laurie and his father in law Mark Buchanan are waiting for him. They are looking for a quick lunch. They might want to go shopping. They might look for some amusement park.

In this scenario, the Situation Provider's understanding is that "Dawson Laurie" and "Mark Buchanan" are "relatives", and "Tokyo station" is a "Public place" because "Tokyo Station" is a sub-concept of "Railway_station" which is a sub-concept of "Public_place", and the time "12:00" is a "leisure_hour". The user's situation is determined to be "Family_meeting" by situational reasoning. Task Selector selects only those tasks associated with "Family_meeting" from task knowledge base and sends those to the Task Navigator on the mobile handset. The user can select tasks displayed on the mobile phones (Fig. 6 center), "Go to movie theatre near station", "Go shopping near station", "Go to amusement park near station", and operate the task structure related to the selected task (Fig. 6 right).



Task Navigator
Who is there with you?
Fiona Davidson
Gordon Green
Mark Buchanan
Campbell Laurie
Where are you now?
Tokyo station
Dawson's home
Current time?
12:00
18:00
send
TOP

Task Navigator
Recommended task
PLACE: STATION
SITUATION: FAMILY
◆Go to movie theater near station
◆Go shopping near station
◆Go to amusement park near station
Other tasks at station
Back

Task Navigator
What do you want to do?
⊟Watch movie(104)
⊟Decide movie title(90)
Check movie schedule(80)
Check movie story(40)
Check movie evaluation(28)
Check movie ranking(9)
Book theater seat(s)(6)
⊞Decide movie theater(11)
戻る

Fig. 6. Flow of service recommendation for meeting family and relative at station

5 Conclusion and Future Work

This paper proposed a task-oriented service recommendation system based on situational reasoning for mobile users that aims at easing access to services which are most appropriate for achieving the user's task in a given real world situations. The system features a task knowledge base that contains knowledge about which services will solve the problems that a user faces in daily life and a situation provider that determines the user's situation based on multiple context information such as location, time and company.

We plant to advance the prototype towards acquiring more actual context information from the real world. Planed extensions to the situation provider will combine GPS-based location information and RFID-based context tags in the user's environment for location tracking as well as or short distance wireless communication technologies such as Bluetooth to detect people in proximity.

References

1. NTT DoCoMo web site.: http://www.nttdocomo.com/
2. Naganuma, T., Kurakake, S.: Task Knowledge Based Retrieval for Service Relevant to Mobile User's Activity, 4th international semantic web conference (ISWC'05) (2005) 959-973
3. Deborah L. McGuinness and Frank van Harmelen: OWL Web Ontology Language Overview. W3C Recommendation (2004). <http://www.w3.org/TR/owl-features/>.
4. David L. Martin, Massimo Paolucci, Sheila A. McIlraith, et al.: Bringing Semantics to Web Services: The OWL-S Approach. First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC'04), San Diego, California, USA (2004) 26-42
5. Patrik Floreen, Michael Przybilski, Petteri Nurmi, Johan Koolwaaij, Anthony Tarlano, Matthias Wagner, Marko Luther, Fabien Bataille, Mathieu Boussard, Bernd Mrohs, and SianLun Lau. Towards a context management framework for MobiLife. In 14th IST Mobile and Wireless Communication Summit (MOWICOM'05), Dresden, Germany (2005)
6. Sebastian Bohm, Marko Luther, and Matthias Wagner. Smarter groups - reasoning on qualitative information from your desktop. In Proceedings of the 1st Workshop on The Semantic Desktop at the ISWC'05 (CEUR-WS Vol 175), Galway, Ireland (2005)
7. Bernd Mrohs, Marko Luther, and Raju Vaidya: Context-aware presence management. In Proceedings of the Workshop on Context Awareness for Proactive Systems (CAPS'05), Helsinki, Finland (2005) 177-180
8. Marko Luther, Sebastian Bohm, Matthias Wagner, and Johan Koolwaaij: Enhanced presence tracking for mobile applications. In Proceedings of the Demo Track of the 4th International Semantic Web Conference (ISWC'05), Galway, Ireland (2005)
9. Marko Luther, Bernd Mrohs, Matthias Wagner et al.: Situational reasoning - a practical OWL use case. In Proceedings of the 7th International Symposium on Autonomous Decentralized Systems (ISADS'05), Chengdu, China (2005)

Author Index