Industrial Applications of Semantic Web

Edited by
Max Bramer
Vagan Terziyan

Springer

ifip

INDUSTRIAL APPLICATIONS OF SEMANTIC WEB

IFIP – The International Federation for Information Processing

IFIP was founded in 1960 under the auspices of UNESCO, following the First World Computer Congress held in Paris the previous year. An umbrella organization for societies working in information processing, IFIP's aim is two-fold: to support information processing within its member countries and to encourage technology transfer to developing nations. As its mission statement clearly states,

> *IFIP's mission is to be the leading, truly international, apolitical organization which encourages and assists in the development, exploitation and application of information technology for the benefit of all people.*

IFIP is a non-profitmaking organization, run almost solely by 2500 volunteers. It operates through a number of technical committees, which organize events and publications. IFIP's events range from an international congress to local seminars, but the most important are:

- The IFIP World Computer Congress, held every second year;
- Open conferences;
- Working conferences.

The flagship event is the IFIP World Computer Congress, at which both invited and contributed papers are presented. Contributed papers are rigorously refereed and the rejection rate is high.

As with the Congress, participation in the open conferences is open to all and papers may be invited or submitted. Again, submitted papers are stringently refereed.

The working conferences are structured differently. They are usually run by a working group and attendance is small and by invitation only. Their purpose is to create an atmosphere conducive to innovation and development. Refereeing is less rigorous and papers are subjected to extensive group discussion.

Publications arising from IFIP events vary. The papers presented at the IFIP World Computer Congress and at open conferences are published as conference proceedings, while the results of the working conferences are often published as collections of selected and edited papers.

Any national society whose primary activity is in information may apply to become a full member of IFIP, although full membership is restricted to one society per country. Full members are entitled to vote at the annual General Assembly, National societies preferring a less committed involvement may apply for associate or corresponding membership. Associate members enjoy the same benefits as full members, but without voting rights. Corresponding members are not represented in IFIP bodies. Affiliated membership is open to non-national societies, and individual and honorary membership schemes are also offered.

INDUSTRIAL APPLICATIONS OF SEMANTIC WEB

Proceedings of the 1st IFIP WG12.5 Working Conference on Industrial Applications of Semantic Web, August 25-27, 2005, Jyväskylä, Finland

Edited by

Max Bramer
University of Portsmouth
United Kingdom

Vagan Terziyan
University of Jyväskylä
Finalnd

Springer

Library of Congress Cataloging-in-Publication Data

A C.I.P. Catalogue record for this book is available from the Library of Congress.

Industrial Applications of Semantic Web, Edited by Max Bramer and Vagan Terziyan

p.cm. (The International Federation for Information Processing)

ISBN-10: (HB) 0-387-28568-7
ISBN-13: (HB) 978-0387-28568-9
Printed on acid-free paper.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1 SPIN 11545873 (HC)
springeronline.com

Contents

Foreword

The Semantic Web, that adds a conceptual layer of machine-understandable metadata to the existing content, will make the content available for processing by intelligent software allowing automatic resource integration and providing interoperability between heterogeneous systems. The Semantic Web is now the most important influence on the development of the Web. Next generation of intelligent applications will be capable to make use of such metadata to perform resource discovery and integration based on its semantics. Semantic Web, aims at developing a global environment on top of Web with interoperable heterogeneous applications, agents, web services, data repositories, humans, and so on. On the technology side, Web-oriented languages and technologies are being developed (e.g. RDF, OWL, OWL-S, WSMO, etc.), and the success of the Semantic Web will depend on a widespread industrial adoption of these technologies. Trend within worldwide activities related to Semantic Web definitely shows that the technology has emerging growth of interest both academic and industry during a relatively small time interval.

VAGAN TERZIYAN
University of Jyväskylä

Organization

IASW-2005 was organized by Industrial Ontologies Group within University of Jyväskylä in cooperation with International Federation for Information Processing (IFIP), Technical Committee 12 (Artificial Intelligence), Working Group 12.5 (Artificial Intelligence Applications) and Knowledge Web Network of Excellence.

Program Committee

Terziyan, Vagan (Co-Chair), University of Jyväskylä, Finland

Bramer, Max (Co-Chair), University of Portsmouth, UK

Bergenti, Federico, FrameTech S.R.L., Italy

Berger, Michael, Siemens AG, Germany

Binder, Walter, EPFL, Switzerland

Denker, Grit, SRI International, USA

Devedzic, Vladan, University of Belgrade, Serbia and Montenegro

Dourgnon-Hanoune, Anne, EDF R&D (Électricité de France), France

Ermolayev, Vadim, Zaporozye State University, Ukraine

Fensel, Dieter, DERI, Innsbruck, Austria

Fernández, Alberto, URJC, Spain

García-Serrano, Ana, Technical University of Madrid, Spain

Helin, Heikki, TeliaSonera, Helsinki, Finland

Karpouzis, Kostas, National Tech. Univ. of Athens, Greece

Klusch Matthias, DFKI, Germany

Laukkanen, Mikko, TeliaSonera, Helsinki, Finland

Lytras, Miltiadis, ELTRUN Research Center, Greece

Mercier-Laurent, Eunika, EML Conseil Knowledge Management, France

Navrat, Pavol, University of Technology in Bratislava, Slovakia

Nykanen, Ossi, W3C Finland, Tampere, Finland

Poslad, Stefan, Queen Mary University of London, UK

Pyotsiä, Jouni, Metso Automation, Helsinki, Finland

Salminen, Airi, University of Jyväskylä, Finland

Schuldt, Heiko, UMIT, Austria

Shoumen, Datta, MIT, USA

Seilonen, Ilkka, Helsinki University of Technology, Finland

Stuckenschmidt, Heiner, Free University of Amsterdam, The Netherlands

Studer, Rudi, AIFB, University of Karlsruhe, Germany

Szeredi, Peter, Univ. of Technology and Economics, Hungary

Toivonen, Santtu, VTT Information Technology, Finland

Tsalgatidou, Aphrodite, University of Athens, Greece

Local Organizing Committee

Olena Kaykova, Industrial Ontologies Group, Univ. of Jyväskylä

Oleksiy Khriyenko, Industrial Ontologies Group,Univ. of Jyväskylä

Pekka Neittaanmäki, Agora Center, Univ. of Jyväskylä

Airi Salminen, CS&IS Dept., Univ. of Jyväskylä

Timo Tiihonen, MIT Dept, Univ. of Jyväskylä

Jari Veijalainen, ITRI, Univ. of Jyväskylä

Andriy Zharko, Industrial Ontologies Group, Univ. of Jyväskylä

Conference Partners and Sponsors

International Federation for Information Processing, Technical Committee TC12 (Artificial Intelligence), Working Group 12.5 (Artificial Intelligence Applications)

Metso Corporation

TeliaSonera Corporation

Knowledge Web Network of Excellence

Conference Management

P.-L. Pitkänen, Jyväskylä Congresses, Finland

Preface

The main focus of the First International IFIP/WG12.5 Working Conference on Industrial Applications of Semantic Web, IASW-2005, held in Jyväskylä, Finland, August 25-27, 2005, is related to industrial applications of Semantic Web. The three more specific concerns within the focus are as follows:

The growing interest to the Semantic Web, as a research and educational domain, from the academy is evident. New scientific results and interesting challenges in the area appear rapidly. International networks cover topics related to intersections of various former scientific domains with Semantic Web technology and discover new challenging opportunities. Basic standards have been announced and the amount of pilot tools and applications around these standards is exponentially increasing. The question is how much the researchers are taking into account the applicability of their results to the industry? The Conference concerns to collect cases from scientists about industrial implementatiuon of their Semantic Web related solutions or to hear arguments in favor of possibilities for such implementation.

In spite of growing hype around Semantic Web and appropriate standards, industry developed and is continuously developing own standards for interoperability and integration. What are the obstacles, companies will face or reasons for refusing wider scale implementation of Semantic Web standards? The Conference aims to collect grounded critics and doubts, related to Semantic Web standards and activities, from industry to raise open discussion between industry and academy concerning future of industrial approval of the Semantic Web technology.

On the other hand, more and more companies are being involved to various projects related to Semantic Web. Industrial investments to research projects aimed to monitor the status of the technology are also growing. Some companies are extensively involved to the appropriate business. There are at least two categories of such enterprises: those who are producers and providers of Semantic Web based products and services and those who are consumers of these products and services. It would be interesting to hear an answer to the question "Why?" they are doing this. We are encouraging representatives from industry to present their opinions about feasibility of Semantic Web technology for their businesses. The Conference aims to collect grounded optimistic arguments, cases and success-stories from such companies.

VAGAN TERZIYAN
University of Jyväskylä

Acknowledgements

The idea of the conference would never emerge, if it was not for the latest project of Industrial Ontologies Group - SmartResource project. We would like to express our deep appreciation to our project partners – Heikki Helin, Jouni Pyötsiä. Many thanks to IFIP for encouragement, organizational and technical support, Prof. John Debenham for his feedback at final stages of the manuscript preparation. Our appreciation is also extended to the PC members for collaboration within tight timetable and their quality paper selection. Our kind thanks are dedicated to Springer for their smooth and professional support. Finally, we are very grateful to persons who have contributed so much to the whole process of the conference organization: Olena Kaykova, Timo Tiihonen, Andriy Zharko and Oleksiy Khriyenko.

PART 1
INVITED KEY-NOTE TALKS

USING THE SEMANTIC WEB IN MOBILE AND UBIQUITOUS COMPUTING

Ora Lassila
Nokia Research Center
Burlington, MA, USA
ora.lassila@nokia.com

Abstract This paper views the Semantic Web as a means to improve the interoperability between systems, applications, and information sources. Emerging personal computing paradigms such as ubiquitous and mobile computing will benefit from better interoperability, as this is an enabler for a higher degree of automation of many tasks that would otherwise require the end-users' attention. Specific application areas of Semantic Web technologies with direct ramifications to these new paradigms include Web Services, context-awareness and policy modeling.

Keywords: Semantic Web, Ubiquitous Computing, Mobile Computing, Context-Awareness, Interoperability

1. Introduction

The Semantic Web [Berners-Lee et al., 2001] – motivated by the rapidly growing volume of useful electronically accessible information that is only meaningful with *human interpretation* – is an effort to build more "machine-friendly" content (and *services*) for the World Wide Web. Information with *accessible formal semantics* can be processed by automated systems (such as *autonomous agents*) without human intervention or the need to apply human interpretation (which we should consider a scarce, critical resource). Deployment of the Semantic Web could ease the current human workload if it leads to easier automation of (Web-based) tasks and thus allows computers to do more *on behalf of* humans.

Much of the promise of the Semantic Web is predicated on the emergence of *ontologies*, specifications of conceptualization [Gruber, 1993] that – in essence – establish "meaning" by defining the relationships between terms of discourse and enabling *reasoning* as a key process through which implicit information can be uncovered.

The emergence of standards for the Semantic Web has made it possible to start deploying the associated technologies outside research laboratory settings. Most importantly, RDF [Lassila, 1998, Lassila and Swick, 1999, Brickley and Guha, 2003] as a formalism for expressing simple taxonomical *ontologies* and OWL [McGuinness and van Harmelen, 2004] for more "expressive" ontologies now form the cornerstone of future Semantic Web development. These are followed by further developments for languages that allow the expression of *queries* [Prud'hommeaux and Seaborne, 2005] and *rules* [Horrocks et al., 2004].

This paper discusses the possible application of Semantic Web technologies to two new paradigms of personal computing, namely *ubiquitous* and *mobile* computing. This application is motivated by the need for better *automation of user's tasks* (as a means of making the user's life easier); we will adopt the view that automation is best enabled by improving the *interoperability* between systems, applications, and information.

2. Enabling Interoperability

To fully realize the vision of the Semantic Web, we must not only address representational issues but also tackle behavioral ones. *Serendipitous interoperability* [Lassila, 2002] – that is, the unarchitected, unanticipated encounters of agents on the Web – is an important component of this realization. Semantic Web techniques – applying knowledge representation techniques in a distributed environment – have proven useful in providing richer descriptions for Web resources. *Semantic Web Services*, a new research paradigm, is generally defined as the augmentation of Web Service descriptions through semantic annotations, to facilitate the higher automation of service discovery, composition, invocation and monitoring in an open, unregulated and often chaotic environment [Payne and Lassila, 2004].

Just as the success of the deployment of the Semantic Web will largely depend on whether useful ontologies will emerge, so will the Semantic Web Services benefit from mechanisms that allow shared agreements about vocabularies for knowledge representation. Sharing vocabularies allows *automated* interoperability; given a base ontology shared by agents, each agent can extend this ontology while achieving partial understanding of the others; this is analogous to object-oriented programming systems, where a base class defines "common" functionality.

Several activities around Semantic Web Services have emerged, the best known being the OWL-S ontology work originated in DARPA's DAML research program [Ankolekar et al., 2002, Martin et al., 2004].

Semantic Web Services represent an important step toward the full-blown vision of the Semantic Web, in terms of utilizing, managing and creating semantic markup.

The relationship between the Semantic Web and the current Web Service architecture depends on one's viewpoint. In the near term, the deployment of Web Services is critical; here, Semantic Web techniques can enhance the current service architecture. In the longer term, assuming the adoption of the Semantic Web vision, the deployment of Semantic Web techniques will be critical; then, Web Services will offer a ubiquitous infrastructure on which to build the next generation of inter-organizational multi-agent systems.

It is important to note that the Semantic Web represents a potential for *qualitatively stronger* interoperability than the traditional standards-based approach. With the latter, one essentially has to anticipate all future scenarios, whereas in the Semantic Web approach it is possible for agents to "learn" new vocabularies and – via reasoning – make meaningful use of them. Furthermore, in addition to current notions of device and application interoperability, the Semantic Web represents interoperability at *the level of the information itself.*

3. Semantic Web Meets Ubiquitous Computing

Ubiquitous computing is an emerging paradigm for personal computing and communications [Weiser, 1991]. Although much of ubiquitous computing research has focused on user interface aspects [Abowd and Mynatt, 2000], we can argue that a characteristic of the paradigm – and thereby distinctly different from the current personal computing paradigm – is the proliferation of devices that need to be connected. Today's user connects his PC to a handful of other devices (printers, network gateways, etc.) and these connections are fairly static, but we anticipate ubiquitous computing scenarios to involve dozens, if not hundreds of devices (sensors, external input and output devices, remotely controlled appliances, etc.). Furthermore, with the advent of mobility and associated proximity networking, the set of connected devices will constantly change as the usage context changes and as devices come into and leave the range of the user's ubiquitous computing device(s). Because of the dynamic nature of the new paradigm, technologies that improve interoperability will be crucial.

Given the need to dynamically connect to a large ever-changing set of devices and services, devices in a ubiquitous computing environment should be capable of sophisticated discovery and *device coalition formation*: the goal should be to accomplish discovery and configuration of new devices without "a human in the loop." In other words, the ultimate objective is the discovery and utilization of services offered by other automated systems without human guidance or intervention, thus enabling the automatic formation of device coalitions through this mechanism. Semantic Web Services, because of the benefits enabled by the application of ontological techniques (as described earlier), appears to be an appropriate paradigm to be applied in representing the functionality of ubiquitous computing devices. Virtual and physical functions can be abstracted as Web Services, providing a uniform view of all different kinds of functionality [Lassila and Adler, 2003, Masuoka et al., 2003]. Again, realization of this is contingent on the continuing emergence of suitable ontologies for modeling ubiquitous computing environments [Chen et al., 2004].

Avoiding *a priori* commitments about how devices are to interact with one another will improve interoperability and thus will make dynamic, unchoreographed ubiquitous computing scenarios more realistic. With reference to the aforementioned *serendipitous interoperability*, the true fulfillment of the vision for ubiquitous computing has a promise of serendipity in it that cannot be realized without discovery mechanisms that are qualitatively stronger than the current practice.

4. Towards Mobile Information Access

The advent of *smartphones* – mobile phones capable of functions typically associated with personal digital assistants (PDAs) or even personal computers – has made *mobile information access* an everyday reality. Although smartphones still suffer from various technical limitations compared to, say, laptop computers (e.g., smaller screen, slower network connectivity, often awkward keyboard input), progress is being made to make Web browsing a typical task on these devices [Nokia, 2005].

Eventually, the physical limitations inherent in mobile information access can be overcome, but we believe that the real limitations have more to do with the usage situations of mobile devices. Information access often (if not predominantly) takes place in situations where the user is "attention-constrained"; in other words, the user is primarily paying attention to *something else* (say, driving a car) and cannot expend full attention to the process of finding and retrieving information. Given that her attention is focused elsewhere, the mobile user may merely "have questions" and will need very specific (and thus potentially terse) answers.

A number of techniques can be applied to help focus the search and acquisition of information. For example, *context awareness* – the identification of usage situations and user's tasks, and tailoring the system behavior based on these [Dey et al., 2001] – can be used to narrow the scope of user's requests. Semantic Web technologies (knowledge representation, reasoning, and the interchange of representations) are well suited to representing and processing context information [Lassila and Khushraj, 2005]. Determining context, however, typically benefits from access to as many sources of information as possible (related to the user, her task, the environment, etc.), and without a proper solution for security, privacy and trust, efforts to implement context-awareness may be hampered. Fortunately Semantic Web techniques are also well suited to describing, reasoning about, and exchanging *policies* which can be used to represent these [Kagal et al., 2003, Kagal, 2004]; this naturally applies to ubiquitous computing environments as well.

Generally, having access to information in "raw" form (i.e., without any forethought as to how the information is to be presented or formatted), combined with the representation and reasoning capabilities enabled by Semantic Web technologies, will be helpful, because then *what* information gets presented and *how* it gets formatted can be a context-based decision. We can think of *context* very broadly, covering just about everything that is known about the user, her task, the current environment, and the *device* she is using to access information. In this regard, it may be possible to go well beyond contemporary *content repurposing* approaches (such as [Nokia, 2003]). For example, it is possible to demonstrate that Semantic Web techniques can be used not only to automatically generate user interfaces from OWL-S descriptions, but that these user interfaces can be contextually optimized for small-screen devices [Khushraj and Lassila, 2005].

5. Conclusions

Semantic Web technologies offer several benefits to new computing paradigms such as mobile and ubiquitous computing. Not only do Semantic Web technologies lend themselves well to representation, reasoning and exchange of many different kinds of information (such as functionality, contexts, policies, user models, etc.), but generally these technologies are a qualitatively stronger approach to *interoperability* than contemporary standards-based approaches. With sophisticated ontological representations we can realize effortless access to heterogeneous information sources, independent of the device being used or the user's context; furthermore, we can finally untap the serendipitous potential that exists in unchoreographed encounters of automated and autonomous systems in cyberspace.

References

[Abowd and Mynatt, 2000] Abowd, G. D. and Mynatt, E. D. (2000). Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction*, 7(1):29–58.

[Ankolekar et al., 2002] Ankolekar, A., Burstein, M., Hobbs, J. R., Lassila, O., McDermott, D., Martin, D., McIllraith, S. A., Narayanan, S., Paolucci, M., Payne, T., and Sycara, K. (2002). DAML-S: Web Service Description for the Semantic Web. In Horrocks, I. and Hendler, J., editors, *The Semantic Web - ISWC 2002*, Lecture Notes in Computer Science 2342, pages 348–363. Springer Verlag.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5):34–43.

[Brickley and Guha, 2003] Brickley, D. and Guha, R. (2003). RDF Vocabulary Description Language 1.0: RDF Schema. W3C Working Draft, World Wide Web Consortium.

[Chen et al., 2004] Chen, H., Perich, F., Finin, T., and Joshi, A. (2004). SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In *International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous)*, Boston, MA.

[Dey et al., 2001] Dey, A., Abowd, G., and Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16:97–166.

[Gruber, 1993] Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220.

[Horrocks et al., 2004] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., and Dean, M. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission, World Wide Web Consortium.

[Kagal, 2004] Kagal, L. (2004). *A Policy-Based Approach to Governing Autonomous Behavior in Distributed Environments.* PhD thesis, University of Maryland Baltimore County, Baltimore MD.

[Kagal et al., 2003] Kagal, L., Parker, J., Chen, H., Joshi, A., and Finin, T. (2003). *Security, Privacy and Trust in Mobile Computing Environments*, chapter Security and Privacy Aspects. CRC Press.

[Khushraj and Lassila, 2005] Khushraj, D. and Lassila, O. (2005). Ontological Approach to Generating Personalized User Interfaces for Web Services. To appear.

[Lassila, 1998] Lassila, O. (1998). Web Metadata: A Matter of Semantics. *IEEE Internet Computing*, 2(4):30–37.

[Lassila, 2002] Lassila, O. (2002). Serendipitous Interoperability. In Eero Hyvönen, editor, *The Semantic Web Kick-off in Finland – Vision, Technologies, Research, and Applications*, HIIT Publications 2002-001. University of Helsinki.

[Lassila and Adler, 2003] Lassila, O. and Adler, M. (2003). Semantic Gadgets: Ubiquitous Computing Meets the Semantic Web. In Fensel, D., Hendler, J., Wahlster, W., and Lieberman, H., editors, *Spinning the Semantic Web*, pages 363–376. MIT

[Lassila and Khushraj, 2005] Lassila, O. and Khushraj, D. (2005). Contextualizing Applications via Semantic Middleware. Accepted to The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous).

[Lassila and Swick, 1999] Lassila, O. and Swick, R. R. (1999). Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, World Wide Web Consortium.

[Martin et al., 2004] Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., and Sycara, K. (2004). OWL-S: Semantic Markup for Web Services. W3C Member Submission, World Wide Web Consortium.

[Masuoka et al., 2003] Masuoka, R., Parsia, B., and Labrou, Y. (2003). Task Computing – the Semantic Web meets Pervasive Computing. In *Proceedings of the 2nd International Semantic Web Conference 2003 (ISWC 2003).*

[McGuinness and van Harmelen, 2004] McGuinness, D. L. and van Harmelen, F. (February 2004). OWL Web Ontology Language Overview. W3C Recommendation, World Wide Web Consortium.

[Nokia, 2003] Nokia (2003). Nokia acquires Eizel to enhance mobile enterprise portfolio. Nokia press release, 2003-04-22.

[Nokia, 2005] Nokia (2005). Nokia develops a new browser for Series 60 by using open source software. Nokia press release, 2005-06-13.

[Payne and Lassila, 2004] Payne, T. and Lassila, O. (2004). Semantic Web Services (guest editors' introduction). *IEEE Intelligent Systems*, 19(4):14–15.

[Prud'hommeaux and Seaborne, 2005] Prud'hommeaux, E. and Seaborne, A. (2005). SPARQL Query Language for RDF. W3C Working Draft, World Wide Web Consortium.

[Weiser, 1991] Weiser, M. (1991). The computer for the twenty-first century. *Scientific American*, 265(3):94–104.

Acknowledgements

The author's own research cited in this paper was supported in part by the Nokia Technology Platforms and the Nokia Research Center.

Semantic Web applications: Fields and Business cases. The Industry challenges the research.

Alain Léger[1], Lyndon J.B. Nixon[2], Pavel Shvaiko[3], Jean Charlet[4],

[1] France Telecom R&D - Rennes, 4 rue du clos courtel, Boîte postale 91226, 35512 Cesson-Sévigné, France
alain.leger@rd.francetelecom.com
[2] Freie Universität Berlin, Takustrasse, 9
14195 Berlin, Germany
nixon@mi.fu-berlin.de
[3] University of Trento (UniTn)
Via Sommarive 14
38050 Trento, Italy
pavel@dit.unitn.it
[4] Jean Charlet, STIM, DPA/AP-Hopitaux Paris & Université Paris 6, 75006 Paris, France
charlet@biomath.jussieu.fr

Abstract. Semantic web technology is more and more often applied to a large spectrum of applications where domain knowledge is conceptualized and formalized (Ontology) as a support for diversified processing (Reasoning) operated by machines. Moreover through a subtle joining of human reasoning (cognitive) and mechanical reasoning (logic-based), it is possible for humans and machines to share complementary tasks. To name few of those applications areas: Corporate Portals and Knowledge Management, E-Commerce, E-Work, Healthcare, E-Government, Natural Language understanding and Automated Translation, Information search, Data and Services Integration, Social networks and collaborative filtering, Knowledge Mining, etc. From a social and economic perspective, this emerging technology should contribute to growth in economic wealth, but it must also show clear cut value in our everyday activities in being technology transparent and efficient. The uptake of Semantic Web technology by industry is progressing slowly. One of the problems is that academia is not always aware of the concrete problems that arise in industry. Conversely, industry is not often well informed about the academic developments that can potentially meet its needs. In this paper we present an ongoing work in the cross-fertilization between industry and academy. In particular, we present a collection of applications fields and use cases from enterprises which are interested in the promises of Semantic Web technology. We explain our approach in the analysis of the industry needs. We summarize industrial knowledge processing requirements in the form of a typology of knowledge processing tasks. These results are intended to focus academia on the development of plausible knowledge-based solutions for concrete industrial problems, and therefore, facilitate the uptake of Semantic Web technology within industry.

1 Introduction

Through the invading, pervasive and user-friendly digital technology within the information society, the fully open web content emerges as multiform, inconsistent and very dynamic. This situation leads to abstracting (via Ontology) this complexity and to offer new and enriched services able to reason on those abstractions (Reasoning) via automata – e.g. Web services. This abstraction layer is the subject of a very dynamic activity in research, industry and standardization in what is usually called worldwide "Semantic Web" [e.g. DARPA, European IST Research Framework Program, W3C]. The very first application of the semantic web technology has focused on Information Retrieval (IR) where access by to semantic content instead of the classical (even sophisticated) statistical analysis was sought to give far better results (Precision and Recall). The next natural extension was on IR applied to enterprise legacy databases integration for leveraging the company information silos. The present large field of applications is now focusing on the seamless integration of applications or services through a full usage of semantic web services for expected fast ROI and efficiency in E-Work and E-Business.

This new technology takes its roots in the cognitive sciences, machine learning, natural language processing, multi-agents systems, knowledge acquisition, mechanical reasoning, logics and decision theory. It can be separated in two distinct – but cooperating fields - one adopting a formal and algorithmic approach for common sense automated reasoning (automated Web) and the other one "keeping the human being in the loop" for socio-cognitive semantic web (automated social Web).

On a large scale, industry awareness of the knowledge-based technology has started only recently, e.g., at the EC level with the IST-FP5 thematic network Ontoweb[1] which had brought together around 50 motivated companies worldwide.

Based on this experience, within the IST-FP6 network of excellence Knowledge-Web[2], an in-depth analysis of the concrete industry needs in the key economic sectors has been identified as one of the next steps towards stimulating the industrial uptake of Semantic Web technology.

The paper is organized as follows. Three prototypical application fields are presented in Section 2: KM, E-Commerce and Healthcare. Use cases collection methodology from industry and their preliminary analysis leading to the identification of key knowledge processing components are presented in Section 3. Finally, Section 4 reports some conclusions and discusses future effort.

---

[1] http://www.ontoweb.org

[2] http://knowledgeweb.semanticweb.org

2 Some prototypical application fields

2.1 Knowledge Management

Nowadays, knowledge is one of the key success factors for today and tomorrow's enterprises. Therefore, company Knowledge Management (KM) has been identified as a strategic tool for enterprises. However, if Information Technology is one foundation element of KM, KM is also interdisciplinary by nature, and includes human resource management, enterprise organization and culture[3].

So KM is the management of the activities and the process aiming at leveraging the use and the creation of knowledge in organizations for two main objectives: capitalization of the corporate knowledge and durable innovation, and fully aligned with the strategic objectives of the organization:

1. Access, sharing, reuse of knowledge (explicit or implicit, private or collective) ;
2. Creation of new knowledge.

A recent CEN/ISSS[4] project (KM Workshop 2002-2003) has issued a finalized proposal on good practices in KM (September 2003). The project began in October 2002 on KnowledgeBoard[5], which is the European Commission public KM portal, and is supposed to close with a final set of CEN recommendations in fall 2003 entitled "European guide to Good Practice in Knowledge Management".

The European KM Framework is designed to support a common European understanding of KM, to show the value of this emerging approach and help organizations towards its successful implementation. The Framework is based on empirical research and practical experience in this field from all over Europe and the rest of the world. The European KM Framework addresses all relevant elements of a KM solution and serves as a reference basis for all types of organizations, which aim to improve their performance by handling knowledge in a better way.

---

[3] Some definitions:

" Knowledge management is the systematic, explicit, and deliberate building, renewal and application of knowledge to maximize an enterprise's knowledge related effectiveness and returns from its knowledge assets" (Wiig 1997) [1]

"Knowledge management is the process of capturing a company's collective expertise wherever it resides in databases, on paper, or in people's heads and distributing it to wherever it can help produce the biggest payoff" (Hibbard 1997) [2]

"KM is getting the right knowledge to the right people at the right time so they can make the best decision" (Pettrash 1996) [3]

[4] http://www.cenorm.be/cenorm/index.htm

[5] http://www.knowledgeboard.com

2.1.1 Where should Knowledge-based KM benefit?

In the past, Information Technology for knowledge management has focused on the management of knowledge containers using text documents as the main repository and source of knowledge. In the future, Semantic Web technology, especially ontologies and machine-interpretable metadata will pave the way to KM solutions that are based on semantically related pieces of knowledge. The knowledge backbone is made of ontologies that define a shared conceptualization of the application domain at hand and provide the basis for defining metadata, that have precisely defined semantics, and that are therefore machine-interpretable. Although, the first KM approaches and solutions have shown the benefits of ontologies and related methods, a large number of open research issues still exist that have to be addressed in order to make Semantic Web technologies a complete success for KM solutions:

- Industrial KM applications have to avoid any kind of overheads as far as possible. Therefore, a **seamless integration** of knowledge creation, e.g. content and metadata specification, and knowledge access, e.g. querying or browsing, into the working environment is required. Strategies and methods are needed to support the creation of knowledge, as side effects of activities that are carried out anyway. These requirements mean **emergent semantics,** e.g. through **ontology learning,** are needed, which reduces the current time consuming task of building-up and maintaining ontologies.

- Access as well as presentation of knowledge has to be **context-dependent**. Since the context is set-up by the current business task, and thus, by the business process being handled, a tight integration of business process management and knowledge management is required. KM approaches can manage knowledge and provide a promising starting point for **smart push services** that will proactively **deliver relevant knowledge** for carrying out the task at hand more effectively.

- **Conceptualization** has to be supplemented by **personalization**. On one hand, taking into account the experience of the user and his/her personal needs is a prerequisite in order to avoid information overload, and on the other hand **to deliver knowledge on the right level of granularity**.

The development of knowledge portals serving the needs of companies or communities is still more or less a manual process. Ontologies and related metadata provide a promising conceptual basis for generating parts of such knowledge portals. Obviously, among others, conceptual models of the domain, of the users and of the tasks are needed. The **Generation of knowledge portals** has to be supplemented with the (semi-) automated evolution of portals. As business environments and strategies change rather rapidly, **KM portals have to be kept up-to-date in this fast changing environment.** Evolution of portals should also include some mechanism **to 'forget' outdated knowledge**.

KM solutions will be based on a combination of intranet-based functionalities and mobile functionalities in the very near future. Semantic Web technologies are a promising approach to meet the needs of mobile environments, like e.g. location-aware personalization and adaptation of the presentation to the specific needs of mobile devices, i.e. the presentation of the required information at an appropriate level of granularity. In essence, employees should have access to the KM application **anywhere** and **anytime**.

Peer-to-Peer computing (P2P), combined with Semantic Web technology, will be an interesting of getting rid of the more centralized KM solutions that are currently used in ontology-based solutions. P2P scenarios open up the way to derive consensual conceptualizations among employees within an enterprise in a bottom-up manner.

Virtual organizations are becoming more and more important in business scenarios, mainly due to decentralization and globalization. Obviously, semantic interoperability between different knowledge sources, as well as trust, is necessary in inter-organizational KM applications.

The integration of KM applications (e.g. skill management) with **E-Learning** is an important field that enables a lot of synergy between these two areas. KM solutions and E-Learning must be integrated from both an organizational and an IT point of view. Clearly, interoperability and integration of (metadata) standards are needed to realize such integration.

Knowledge Management is obviously a very promising area for exploiting Semantic Web technology. Document-based KM solutions have already reached their limits, whereas semantic technologies open the way to meet the KM requirements in the future.

2.1.2 Knowledge-based KM applications[6]

In the context of geographical team dispersion, multilingualism and Business Units autonomy, usually the company wants a solution allowing the identification of strategic information, the secured distribution of this information and the creation of transverse working groups. Some applicative solutions allowed the deployment of an Intranet intended for all the marketing departments of the company worldwide, allowing a better division and a greater accessibility to information, but also capitalisation

[6] http://www.arisem.com
http://www.mondeca.com
http://www.ontoknowledge.com
http://www.distributedthinking.com
http://www.ontoknowledge.com
http://www.si.fr.atosorigin.com/sophia/comma/Htm/HomePage.htm

on the total knowledge of the company group. There are three crucial points that aim to ease the work of the various marketing teams of the company group: automatic competitive intelligence of the Web, skill management and document management.

Thus, the system connects the "strategic ontologies" of the company group (brands, competitors, geographical areas, etc...) with the users, via the automation of related processes (research, classification, distribution, representation of knowledge). The result is a dynamic "Semantic Web" system of navigation (research, classification) and collaborative features.

From a functional point of view, KM server organises skill and knowledge management within the company, in order to improve interactivity, collaboration and information sharing. This constitutes a virtual workspace which facilitates work between employees that speak different languages; automates the creation of work groups; organises and capitalises structured and unstructured, explicit or tacit data of the company organisation, and offers advanced features of capitalisation. Furthermore, the semantic backbone also makes possible to cross a qualitative gap by providing cross-lingual data. Indeed, the semantic approach allows ontologies to overcome language barriers (Culture and Language differences).

Some lessons learnt[7]:

- **Main strong benefits** for the enterprise are high productivity gains and operational valorisation of knowledge legacy
- **Productivity**: Automation of knowledge base maintenance, Automation of content indexing, Augmented productivity in publication cycle (commercial proposals, reports ...), Search efficiency (a reduction factor on research time of the order (1000 to 1) is claimed possible by the use of ontologies)
- **Quality and operational valorisation of knowledge legacy**: Unified management of heterogeneous resources, Information relevancy, Capacity to represent complex knowledge, Gains in development and maintenance of knowledge and content management solution, Generic and evolvable solution
- **Human factors** are key difficulties in full groupware functionalities of the KM solution towards the employees of the company, so adopt a step-by-step approach
- **Access to information portal** must be well designed and must be supported by a group of people dedicated to information filtering and qualifying (P2P is possible)

[7] Le Monde Informatique 11 July 2003 and http://www.mondeca.com

2.2 E-Commerce

Electronic Commerce is mainly based on the exchange of information between involved stakeholders using a telecommunication infrastructure. There are two main scenarios: Business-to-Customer (B2C) and Business-to-Business (B2B).

B2C applications enable service providers to promote their offers, and for customers to find offers, which match their demands. By providing a single access to a large collection of frequently updated offers and customers, an electronic marketplace can match the demand and supply processes within a commercial mediation environment.

B2B applications have a long history of using electronic messaging to exchange information related to services previously agreed among two or more businesses. Early plain-text telex communication systems were followed by electronic data interchange (EDI) systems based on terse, highly codified, well structured, messages. A new generation of B2B systems is being developed under the ebXML (electronic business in XML) label. These will use classification schemes to identify the context in which messages have been, or should be, exchanged. They will also introduce new techniques for the formal recording of business processes, and for the linking of business processes through the exchange of well-structured business messages. ebXML will also develop techniques that will allow businesses to identify new suppliers through the use of registries that allow users to identify which services a supplier can offer. ebXML needs to include well managed multilingual ontologies that can be used to help users to match needs expressed in their own language with those expressed in the service providers language(s).

2.2.1 Where is the value of Knowledge-based E-Commerce?

At the present time, ontology and more generally ontology-based systems, appear as a **central issue** for the development of **efficient and profitable** Internet commerce solutions. However, because of an actual lack of standardization for business models, processes, and knowledge architectures, it is currently difficult for companies to achieve the promised ROI from Knowledge-based E-Commerce.

Moreover, a technical barrier exists that delays the emergence of E-Commerce, laying in the need for applications to **meaningfully share information**, taking into account the lack of reliability and security of the Internet. This fact may be explained by the variety of enterprise and e-commerce systems employed by businesses and the various ways these systems are configured and used. As an important remark, such **interoperability problems** become particularly acute when a large number of trading partners attempt to agree and define the standards for interoperation, which is precisely a main condition for maximizing the ROI.

Although it is useful to strive for the adoption of a single common domain-specific standard for content and transactions, such a task is often still difficult to achieve, particularly in cross-industry initiatives, where companies co-operate and compete with one another. Some examples of the difficulties are:

- **Commercial practices** may vary in a wide range and consequently, cannot always be aligned for a variety of technical, practical, organizational and political reasons.
- **The complexity of the global description** of the organizations themselves: their products and services (independently or in combination), and the **interactions** between them remain a formidable task.
- It is usually very difficult to establish, a priori rules (technical or procedural) governing participation in an electronic marketplace.
- Adoption of **a single common standard may limit business models**, which could be adopted by trading partners, and then, potentially reduce their ability to fully participate in Internet commerce.

An ontology based approach has the potential to significantly accelerate the penetration of electronic commerce within vertical industry sectors, by **enabling interoperability at the business level**, reducing the need for standardisation at the technical level. This will enable services to adapt to the rapidly changing online environment.

The following uses for ontologies and classification schemes that could be defined using ontologies, have been noted within electronic commerce applications:

- Categorization of products within catalogues
- Categorization of services (including web services)
- Production of yellow page classifications of companies providing services
- Identification of countries, regions and currencies
- Identification of organizations, persons and legal entities
- Identification of unique products and saleable packages of products
- Identification of transport containers, their type, location, routes and contents
- Classification of industrial output statistics.

2.2.2 Knowledge-based E-Commerce applications

According to (Zyl et al.) [4], applications of this kind use one or more shared ontology to integrate heterogeneous information systems and allow common access for humans or computers. This enforces the shared ontology as the standard ontology for all participating systems, which removes the heterogeneity from the information system. The heterogeneity is a problem because the systems to be integrated are already operational and it is too costly to redevelop them. A linguistic ontology is sometimes used to assist in the generation of the shared ontology, or is used as a top-level ontology, describing very general concepts like space, time, matter, object, event, action, etc, for the shared ontologies to inherit from it. Benefits are the integration of heterogeneous information sources, which can improve interoperability, and more effective use and reuse of knowledge resources[8].

Yellow Pages and products catalogue are direct benefactors of a well structured representation which coupled to multilingual ontology enhances clearly the precision / recall of products or services search engine. The ONTOSEEK system (1996-1998) is the first system being prototyped associating domain ontology (in KR conceptual graph CG with very limited expressiveness) to a large multilingual linguistic ontology (SENSUS – WORDNET) for natural language search of products (Guarino et al., 1998) [5]. ONTOSEEK search products by mapping natural human language human requests to domain ontology. Unlike traditional eCommerce portal search functions the user is not supposed to know the vocabulary used for describing the products and thanks to the SENSUS ontology he is able to express himself in its own vocabulary.

The main functional architectural choice of ONTOSEEK:

- Use of a general linguistic ontology to describe products;
- Great flexibility in expressing the request thanks to the semantic mapping offered between the request and the offers;
- Interactive guided request formulation through generalisation and specialisation links

A Conceptual Graph KR is used internally to represent Request and Products. The semantic matching algorithm is based on a simple subsumption on the ontology graph and does not make use of a complex graph endomorphism.

ONTOSEEK has not been deployed commercially but at its trial period has fully demonstrated the potential benefits making use of preliminary semantic web tools.

The MKBEEM [6] prototype and technology (Multilingual Knowledge Based European Electronic Marketplace - IST-1999-10589, 2000 – 2003) concentrate on written language technologies and its use in the key sector of worldwide commerce. Within the global and multilingual Internet trading environment, there is an increasing pressure on e-content publishers of all types to adapt content for international markets. Localization – translation and cultural adaptation for local markets – is proving to be a key driver of the expansion of business on the web. In particular MKBEEM is focusing on adding multilingualism to all stages of the information cycle, including multilingual content generation and maintenance, automated translation and interpretation, and enhancing the natural interactivity and usability of the service with unconstrained language input. On the Knowledge technology side, the MKBEEM Ontologies provide a consensual representation of the electronic commerce field in two typical Domains (B2C Tourism, B2C Mail order) allowing the commercial exchanges to be transparent in the language of the end user, of the service, or of the product provider. Ontologies are used for classifying and indexing

[8] http://www.chemdex.com
http://kmi.open.ac.uk/projects/alice/
http://www.telecom.ntua.gr/smartec/
http://www.mkbeem.com/

catalogues, for filtering user's query, for selecting relevant products and providers, for facilitating multilingual man-machine dialogues, and for inferring information that is relevant to the user's request and eventually trading needs. The Key Innovative approach is based on a combined use of human language processing and ontologies based reasoning, for:

The effectiveness of the developed generic solutions has been tested in Finnish, French, Spanish and English in the domains of travel booking (SNCF French Rail services) and mail order sales (La Redoute - Ellos).

2.3 Biosciences and Medical applications

The Medical domain is a favourite target for semantic web applications just as the expert system was for Artificial Intelligence applications 20 years ago. The medical domain is effectively very complex: medical knowledge being difficult to represent in a computer, which makes the sharing of information difficult. Semantic web solutions become very promising in this context.

Thus one of the main mechanisms of the semantic web, resource description using annotation principles, is of major importance in the medical informatics (or "bioinformatics") domain, especially as regards the sharing of these resources (e.g. medical knowledge in the Web or genomic data-base). Through the years, the information retrieval domain has been developed by medicine: the medical thesaurus is enormous (1,000,000 terms for UMLS) and is principally used for bibliographic indexation. Nevertheless, the MeSh thesaurus (Medical Subject Heading) or UMLS[9] (Unified Medical Language System) is used in the web semantic paradigm with varying degrees of difficulty. Finally, the web services technology allows us to imagine some solutions to the interoperability problematic, which is substantial in medical informatics. We will describe current research, results and expected perspectives in theses biomedical informatics topics in the context of the semantic web.

2.3.1 Biosciences resources sharing

In the functional genomics domain, it is necessary to have access to several data bases and knowledge bases which are accessible via the web but are heterogeneous in their structure as well as in their terminology. Among such resources, we can cite SWISSPROT[10] where the gene products are annotated by Gene Ontology[11], GenBank[12], etc. In comparing the resources, it is easy to see that they propose the same information in different formats. The XML language, described as the unique common language of these bases proposes as much Document Type Definition (DTD) as resources and does not resolve the interoperability problem.

---

[9] http://www.nlm.nih.gov/research/umls/umlsmain.html
[10] http://us.expasy.org/sprot/
[11] http://obo.sourceforge.net/main.html
[12] http://www.ncbi.nlm.nih.gov/Genbank/index.html

The solution comes from the semantic web with the mediator approach (Wiederhold, 1992) [7] which allows the accessing of different resources with an ontology used as Interlingua pivot. For example, and in another domain than that of genomics, the mediator mechanisms, the NEUROBASE project (Barillot et al., 2003) [8] attempts to federate different neuroimagery information bases situated in different clinical or research areas. The proposal consists of defining an IT architecture that allows the access to and the sharing of experimental results or data treatment methodologies. It would be possible to search in the various data bases for similar results or for images with peculiarities or to perform data mining analysis between several data bases. The mediator of NEUROBASE is tested on decision support systems in epilepsy surgery.

2.3.2 Web services for interoperability

The web services technology can propose some solutions to the interoperability problematic. We describe now a new approach based on "patient envelope" and we conclude on the implementation of this envelope with the web services technologies.

The patient envelope is a proposition of the Electronic Data Interchange for Healthcare group (EDI-Santé[13]) with an active contribution from the ETIAM society[14]. The objective of the work has been to focus on filling the gap between "free" communication, using standard and generic Internet tools, and "totally structured" communication as promoted by CEN[15] or HL7[16]. After a worldwide analysis of existing standards, the proposal consists of an "intermediate" structure of information, related to one patient, and storing the minimum amount of data (i.e. exclusively useful data) to facilitate the interoperability between communicating peers. The "free" or the "structured" information is grouped into a folder and transmitted in a secure way over the existing communication networks (Cordonnier et al., 2003) [9]. This proposal has reached widespread celebrity with the distribution by Cegetel.rss of a new medical messaging service, called "Sentinelle", fully supporting the patient envelope protocol and adapted tools.

After this milestone, EDI-Santé is promoting further developments based on ebXML and SOAP (Simple Object Access Protocol) in specifying exchange (1,2) and medical (3, 4) properties:

1. Separate what is mandatory to the transport and the good management of the message (patient identification, ...) from what constitute the "job" part of the message
2. Provide a "container", collecting the different elements, texts, pictures, videos, etc.
3. The patient as unique object of the transaction. Such an exchange cannot be anonymous. It concerns a sender and an addressee who are involved in the ex-

[13] http://www.edisante.org/

[14] http://www.etiam.com/

[15] http://www.centc251.org/

[16] http://www.hl7.org/

change and responsible. The only way to perform this exchange between practitioner about a patient who can demand to know the content of the exchange imply to retain a structure which is unique, a triplet {sender, addressee, patient}.
4. The conservation of the exchange semantics. The information about a patient is multiple. It comes from multiple sources and has multiple forms and supports (data base, free textual document, semi-structured textual document, pictures ...). It can be fundamental to maintain the existing links between elements, to transmit them together, e.g. a scanner and the associated report, and to prove it.

The interest of such an approach is that it prepares the evolution of the transmitted document, from free document (from proprietary ones to normalize as XML) to elements respecting HL7v3 or EHRCOM data types.

2.3.3 And next?

These different projects and applications highlight the main consequence of the semantic web, expected by the medical communities, the sharing and integration of heterogeneous information or knowledge. The answers to the different issues are the mediators, the knowledge-based system, and the ontologies, all based on normalized languages as RDF, OWL or others. The work of the semantic web community must take into account these expectations - see FP6 projects[17,18,19]. Finally, it is interesting to note that the semantic web is an integrated vision of the medical community's problems (thesaurus, ontology, indexation, inference) and provides a real opportunity to synthesize and reactivate some research (Charlet et al., 2002) [10].

3 Use Case collection and Analysis

We have formed a group of companies interested in Semantic Web technology. By the end of 2004, this group consisted of 34 members (e.g., France Telecom, IFP, Illy Caffe, Trenitalia, Daimler Chrysler ...) from across 12 economic sectors (e.g., telecoms, energy, food, logistics, automotive).

The companies were requested to provide illustrative examples of actual or hypothetical deployment of Semantic Web technology in their business settings. This was followed up with face-to-face meetings between researchers and industry experts from the companies to gain additional information about the provided use cases. Thus, in 2004, we collected a total of 16 use cases from 12 companies.

[17] http://www.cocoon-health.com
[18] http://www.srdc.metu.edu.tr/webpage/projects/artemis/index.html
[19] http://www.simdat.org

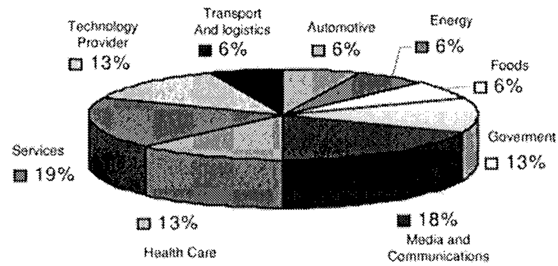


Figure 1: Breakdown of use cases by industry sectors

In particular, it represents (the most active) 9 sectors, with the highest number of the use cases coming from the service industry (19%) and media & communications (18%) respectively. The entire collection of use cases can be found in [11], or on the Outreach to Industry portal[20].

3.1 Preliminary Analysis of Use Cases

A preliminary analysis of the use cases has been carried out in order to obtain a first vision of the current industrial needs and to estimate the expectations from knowledge based technology with respect to those needs. The industry experts were asked to indicate the existing legacy solutions in their use cases, technological locks they encountered, and how they expected that Semantic Web technology could resolve those locks. As a result, we have gained an overview of:

- Types of business problems where the knowledge-based technology is considered to bring a plausible solution;
- Types of technological issues (and corresponding research challenges) which knowledge based technology is expected to overcome.

Let us discuss some concrete types of business problems/technological issues we have identified with the help of experts (see Figure 2 and Figure 3 for a summary).

[20] http://knowledgeweb.semanticweb.org/o2i/

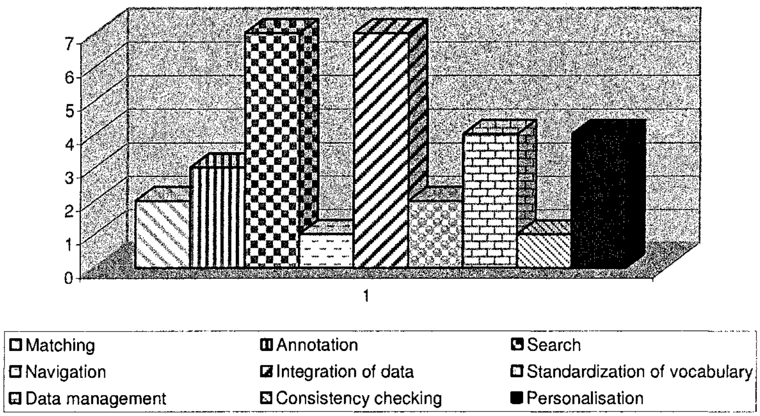


Figure 2: Preliminary vision for solutions sought in use cases

Figure 2 shows a breakdown of the areas in which the industry experts thought Semantic Web technology could provide a solution. For example, for nearly half of the collected use cases data integration and semantic search were areas where industry was looking for knowledge-based solutions. Other areas mentioned, in a quarter of use cases, were solutions to data management and personalization.

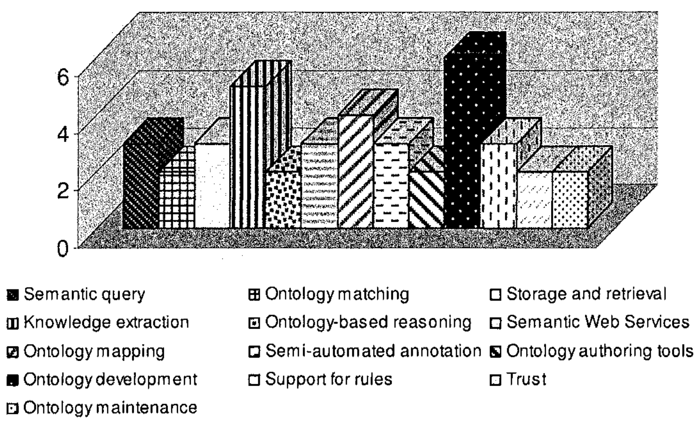


Figure 3: Preliminary vision of technology locks in use cases

Figure 3 shows a breakdown of the technology locks identified in the use cases. There are three technology locks which occur the most often in the collected use cases (namely, from 4 up to 6 use cases). These are: *ontology development*, i.e., modeling of a business domain, authoring, reusing existing ontologies; *knowledge extraction*, i.e., populating ontologies by extracting data from legacy systems; and ontology matching, i.e., resolving semantic heterogeneity among multiple ontologies.

Below, we illustrate, with the help of another use case from our collection, how a concrete business problem can also be used to indicate the technology locks for which knowledge-based solutions potentially might be useful. This use case addresses the problem of an intelligent search of documents in a corporate data of a coffee company.

The company generates large amount of internal data and its employees encounter difficulties in finding the data they need for the research and development of new solutions. The aim is to improve the quality of the documents retrieval and to enable the personalization services of individual users when searching or viewing the corporate data. As technology locks, the expert mentioned here the corporate domain *ontology development* and *maintenance*, and *semantic querying*.

The above three examples illustrate some concrete business scenarios in which an "abstract" research issues such as matching, data integration, etc., are viewed to be of great value to industry. This analysis (by experts estimations) provides us with a preliminary understanding of scope of the current industrial needs and concrete technology locks where knowledge-based technology is expected to provide a plausible solution. However, to be able to answer specific industrial requirements, we need to conduct further a detailed technical analysis of the use cases, thereby associating to each technology lock a concrete knowledge processing task and a component realizing its functionalities.

3.2 Knowledge processing tasks and components

Based on the knowledge processing needs identified during the technical use cases analysis [12], we built a typology of knowledge processing tasks and a library of high level components for realizing those tasks, see Table 1.

N°	Knowledge processing tasks	Components
1	Ontology Management	Ontology Manager
2	Matching	Match Manager
3	Matching results Analysis	Match Manager
4	Data Translation	Wrapper
5	Results Reconciliation	Results Reconciler
6	Composition of Web Services	Planner
7	Content Annotation	Annotation manager
8	Reasoning	Reasoner
9	Semantic Query Processing	Query Processor
10	Schema/Ontology Merging	Ontology Manager
11	Producing explanations	Match Manager
12	Personalization	Profiler

Tablel. Typology of knowledge processing tasks & components

Our first tentative typology includes 12 knowledge processing tasks. Let us discuss knowledge processing tasks and components of Table 1 in more detail.

Ontology Management, Schema/Ontology Merging and Ontology Manager. These tasks and component are in charge of ontology maintenance (e.g., reorganizing taxonomies, resolving name conflicts, browsing ontologies, editing concepts) and merging multiple ontologies (e.g., by taking the union of the axioms) with respect to evolving business case requirements, see [13, 14, 15].

Matching, Matching Results Analysis, Producing Explanations and Match Manager. These tasks and component are in charge of (on-the-fly and semi-automatic) determining semantic mappings between the entities of multiple schemas, classifications, and ontologies, see [16, 17]. Mappings are typically specified with the help of a similarity relation which can be either in the form of a coefficient rating match quality in the [0,1] range (i.e., the higher the coefficient, the higher the similarity between the entities, see [18,19,20,21,22]) or in the form of a logical relation (e.g., equivalence, subsumption), see [23, 24]. The mappings might need to be ordered according to some criteria, see [25, 21].

Finally, explanations of the mappings might be also required, see [26, 27]. Matching systems may produce mappings that may not be intuitively obvious to human users. In order for users to trust the mappings (and thus use them), they need information about them. They need access to the sources that were used to determine semantic correspondences between terms and potentially they need to understand how deductions/ manipulations are performed. The issue here is to present explanations in a simple and clear way to the user.

Data Translation and Wrapper. This task and component is in charge of automatic manipulation (e.g., translation, exchange) of instances between heterogeneous information sources storing their data in different formats (e.g., RDF, SQL DDL, XML), see [28, 29]. Here, mappings are taken as input (for example, from the match manager component) and are analyzed in order to generate query expressions that perform the required manipulations with data instances.

Results Reconciliation and Results Reconciler. This task and component is in charge of determining an optimal solution, in terms of contents (no information duplication, etc.) and routing performance, for returning results from the queried information sources, see [30].

Composition of Web Services and Planner. This task and component is in charge of automated composition of web services into executable processes, see [31]. Composed web services perform new functionalities by interacting with pre-existing services that are published on the Web.

Content Annotation and Annotation Manager. This task and component is in charge of automatic production of metadata for the contents, see [32]. Annotation manager takes as input the (pre-processed) contents and domain knowledge and produces as output a database of content annotations. In addition to the automatic production of content metadata, prompt mechanisms should enable the user with a possibility to enrich the content annotation by adding some extra information (e.g., title, name of a location, title of an event, names of people) that could not be automatically detected.

Reasoning and Reasoner. This task and component is in charge of providing logical reasoning services (e.g., subsumption, concept satisfiability, instance checking tests), see [33]. For example, when dealing with multimedia annotations, logical reasoning can be exploited in order to check consistency of the annotations against the set of spatial (e.g., left, right, above, adjacent, overlaps) and temporal (e.g., before, after, during, co-start, co-end) constraints. Thus, ensuring that the objects detected in the multimedia content correspond semantically to the concepts defined in a domain ontology. For example, in the racing domain, it should be checked whether a car is located above a road or whether the grass and sand are adjacent to the road.

Semantic Query Processing and Query Processor. This task and component is in charge of rewriting a query by using terms which are explicitly specified in the model of a domain knowledge in order to provide a semantics- preserving query answering, see [32, 34]. Examples of queries are "Give me all the games played on grass" or "Give me all the games of double players", in the tennis domain. Finally, users should be able to query by a sample image. In this case, the system should perform an intelligent search of images and videos (e.g., by using semantic annotations) where, for example, the same event or type of activity takes place.

Personalization and Profiler. This task and component is in charge of tailoring services available from the system to the specificity of each user, see [35]. For example, generation and updating of user profiles, recommendation generation, inferring user preferences, and so on. For example users might want to share annotations within trusted user networks, thus having services of personal metadata management and contact's recommender. Also, a particular form of personalization, which is media adaptation, requires knowledge-based technology for a suitable delivery of the contents to the user' terminal (e.g., palm, mobile phone, portable PC).

4. Conclusions and future work

The most relevant initiative to our efforts is IST-FP5 Ontoweb (2001-2004). It formed a special interest group (SIG) on Industrial Applications[21] which collected over 50 use cases. However, the majority of those use cases dealt with technology producers rather than potential adopters of the technology. Ontoweb achieved a good overview of the main roadblocks on the way towards a successful transfer of knowledge-based technology to industry. Based on those foundations, the subsequent IST-FP6 Network of Excellence KnowledgeWeb (2004-2007), has continued the Ontoweb initiative by going into the detail of each particular business case, targeting at (i) collecting industry needs from potential client industry with a specific focus on a few most promising sectors; (ii) identifying the key processing components emerging from the concrete needs analysis; (iii) evaluating research and technology for answering industry needs; (iv) making recommendations through best-of-class guidelines; (v) providing education for practitioners via competence centers, thereby enabling the transfer of a technology know-how.

[21] http://ago.sig4.fr

In this paper we have reported some results on the first two topics as addressed by Knowledge Web. By a preliminary analysis of the collected use cases we categorized the types of solutions being sought for, and the types of technological locks which arise when realizing those solutions. By a detailed technical analysis of the selected use cases we identified precisely where in the business processes the technology locks occur, described the requirements for technological solutions that overcome those locks, and argued for the appropriateness of knowledge-based solutions. Moreover, a quick analysis of the other business cases of [11] have shown that most of the knowledge processing tasks of Table 1 repeat with some variations/specificity from use case to use case. This observation suggests that the constructed typology is stable, i.e., it contains (most of) the core knowledge processing tasks stipulated by the current industry needs. By identifying concrete industry needs through tasks and components, we link them to specific research challenges which we expect the Semantic Web researchers to focus on. As such components are made available from the research, it is possible to evaluate them in different industry-strength settings, and therefore, estimate their practical impact and a contribution to the industrial uptake of Semantic Web technology.

With the emergence of new business cases it is likely that new knowledge processing tasks will appear. For example, web service discovery, orchestration, and so on. Thus, future work includes continuing to collect business cases and to carry out their technical analysis until the saturation is reached.

6. Aknowledgments

The work described in this paper is supported by the EU Network of Excellence Knowledge Web (FP6-507482).

References

1. K.Wiig, Knowledge management: where did it come from and where will it go? Journal of Expert Systems with Applications, 13(1), 1–14, 1997.
2. J. Hibbard, Knowledge management—knowing what we know. Information Week, 20 October 1997.
3. G. Petrash, Managing knowledge assets for value. In Proceedings of the Knowledge-Based Leadership Conference,Boston, MA, October 1996. Boston, MA: Linkage.
4. Zyl J. Corbett D. (2000), A framework for Comparing the use of a Linguistic Ontology in an Application, Workshop Applications of Ontologies and Problem-solving Methods, ECAI'2000, Berlin Germany, August, 2000
5. Guarino N., Masolo C., Vetere G., OntoSeek: (1999) Content-Based Access to the Web, IEEE Intelligent System.
6. MKBEEM (2002) Multilingual Knowledge-Based E-Commerce http://www.mkbeem.com
7. Wiederhold G. (1992). Mediators in the architecture of future information systems, Computer, Vol. 25(3). p.38-49
8. Barillot C., Amsaleg L., Aubry F., Bazin J-P., Benali H., Cointepas Y., Corouge I., Dameron O., Dojat M., Garbay C., Gibaud B., Gros P., Inkingnehun S., Malandain G., Matsumoto J.,

Papadopoulos D., Pélégrini M., Richard N., Simon E., Neurobase: Management of distributed knowledge and data bases in neuroimaging . In Human Brain Mapping, Volume 19, Pages 726-726, New-York, NY, 2003.
9. Cordonnier E., Croci S., Laurent J.-F., Gibaud B. (2003) Interoperability and Medical Communication Using "Patient Envelope"-Based Secure Messaging Proceedings of the Medical Informatics Europe Congress.
10. Charlet J., Cordonnier E., Gibaud B. (2002) Interopérabilité en médecine: quand le contenu interroge le contenant et l'organisation. Revue Information, interaction, intelligence 2(2).
11. L. Nixon, M. Mochol, A. Léger, F. Paulus, L. Rocuet, M. Bonifacio, R. Cuel, M. Jarrar, P. Verheyden, Y. Kompatsiaris, V. Papastathis, S. Dasiopoulou, and A. Gomez Pérez. D1.1.2 Prototypical Business Use Cases. Technical report, Knowledge Web NoE, 2004.
12. P. Shvaiko, A. L'eger, F. Paulus, L. Rocuet, L. Nixon, M. Mochol, Y. Kompatsiaris, V. Papastathis, and S. Dasiopoulou. D1.1.3 Knowledge Processing Requirements Analysis. Technical report, Knowledge Web NoE, 2004.
13. D. Dou, D. McDermott, and P. Qi. Ontology translation on the Semantic Web. Journal on Data Semantics, pages 35–57, 2005.
14. Stanford Medical Informatics. Protégé ontology editor and knowldege aquisition system. http://protege.stanford.edu/index.html.
15. D. L. McGuinness, R. Fikes, J. Rice, and S.Wilder. An environment for merging and testing large ontologies. In Proceedings of KR, pages 483–493, 2000.
16. E. Rahm and P. Bernstein. A survey of approaches to automatic schema matching. VLDB Journal, (10(4)):334–350, 2001.
17. P. Shvaiko and J. Euzenat. A survey of schema-based macthing approaches. Submitted to the Journal on Data Semantics, 2004.
18. A. Billig and K. Sandkuhl. Match-making based on semantic nets: The xml-based approach of baseweb. In Proceedings of the 1st workshop on XML-Technologien fr das Semantic Web, pages 39–51, 2002.
19. M. Ehrig and S. Staab. QOM: Quick ontology mapping. In Proceedings of ISWC, pages 683–697, 2004.
20. J. Euzenat and P.Valtchev. Similarity-based ontology alignment in OWL-lite. In Proceedings of ECAI, pages 333–337, 2004.
21. H.H.Do and E. Rahm. COMA - a system for flexible combination of schema matching approaches. In Proceedings of VLDB, pages 610–621, 2001.
22. J. Zhong, H. Zhu, J. Li, and Y. Yu. Conceptual graph matching for semantic search. In Proceedings of the 2002 International Conference on Computational Science, 2002.
23. F. Giunchiglia and P. Shvaiko. Semantic matching. Knowledge Engineering Review Journal, (18(3)):265–280, 2003.
24. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-Match: an algorithm and an implementation of semantic matching. In Proceedings of ESWS, pages 61–75, 2004.
25. T. Di Noia, E. Di Sciascio, F. M. Donini, and M. Mongiello. A system for principled matchmaking in an electronic marketplace. In Proceedings of WWW, pages 321–330, 2003.
26. R. Dhamankar, Y. Lee, A. Doan, A. Halevy, and P. Domingos. iMAP: Discovering complex semantic matches between database schemas. In Proceedings of SIGMOD, pages 383 – 394, 2004.
27. P. Shvaiko, F. Giunchiglia, P. Pinheiro da Silva, and D. L. McGuinness. Web explanations for semantic heterogeneity discovery. In Proceedings of ESWC, 2005.
28. J. Petrini and T. Risch. Processing queries over rdf views of wrapped relational databases. In Proceedings of the 1st International workshop on Wrapper Techniques for Legacy Systems, Delft, Holland, 2004.
29. Y. Velegrakis, R. J. Miller, and J. Mylopoulos. Representing and querying data transformations. In Proceedings of ICDE, 2005.

30. N. Preguica, M. Shapiro, and C. Matheson. Semantics-based reconciliation for collaborative and mobile environments. In Proccedings of CoopIS, 2003.
31. P. Traverso and M. Pistore. Automated composition of semantic web services into executable processes. In Proceedings of ISWC, pages 380–394, 2004.
32. aceMedia project. Integrating knowledge, semantics and content for user centred intelligent media services. http://www.acemedia.org
33. V. Haarslev, R. Moller, and M. Wessel. RACER: Semantic middleware for industrial projects based on RDF/OWL, a W3C Standard. http://www.sts.tu-harburg.de/~r.f.moeller/racer/
34. E. Mena, V. Kashyap, A. Sheth, and A. Illarramendi. Observer: An approach for query processing in global information systems based on interoperability between pre-existing ontologies. In Proceedings of CoopIS, pages 14–25, 1996.
35. G. Antoniou, M. Baldoni, C. Baroglio, R. Baumgartner, F. Bry, T. Eiter, N. Henze, M. Herzog, W. May, V. Patti, R. Schindlauer, H. Tompits, and S. Schaffert. Reasoning Methods for Personalization on the Semantic Web. Annals of Mathematics, Computing & Teleinformatics,

ENTERPRISE APPLICATIONS OF SEMANTIC WEB: THE SWEET SPOT OF RISK AND COMPLIANCE

Amit Sheth
Semagix, Inc., Athens, GA, USA www.semagix.com

Abstract

Semantic Web is in the transition from vision and research to reality. In this early state, it is important to study the technical capabilities in the context of real-world applications, and how applications built using the Semantic Web technology meet the real market needs. Beyond push from research, it is the market pull and the ability of the technology to meet real business needs that is a key to ultimate success of any technology. In this paper, we discuss the market of Risk and Compliance which presents unique market opportunity combined with challenging technical requirements. We discuss how the Semantic Web technology with an ontology driven approach is especially well suited to support the demanding requirements of the applications in this market. We also discuss the capabilities of a commercial semantic technology that has origins in academic research, as it is utilized in a significant Risk and Compliance application deployed at large financial institutions. Core capabilities of this technology include the ability to develop and maintain focused but large populated ontologies, automatic semantic metadata extraction supported by disambiguation techniques, ability to process heterogeneous information and provide semantic integration combined with link identification and analysis through rule specification and execution, as well as organization and domain specific scoring and ranking. These semantic capabilities are coupled with enterprise software capabilities which are necessary for success of an emerging technology for meeting the needs of demanding enterprise customers.

Keywords: Semantic Web technology, Enterprise Applications, Risk and Compliance, Ontology driven Information Systems, Semantic Metadata, Link Analysis, Rule Processing, Risk Scoring, Customer Identification and Risk Assessment Solution

1. INTRODUCTION

The Semantic Web[1] has arrived. We have early applications that are now functioning and deployed in scientific research as well as industry [Miller2005][Sheth 2005b][Lee 2005]. We also have SW language standards such as RDF and OWL, and we have some stealth applications leading to the pervasive use of enablers such as associating metadata in RDF with digital content over mobile networks/devices and use of metadata in RDF for specifying and validating content license[2]. We also have some early experiences that show where Semantic Web demonstrates clear value and significant differentiation so that we can chart its broader adoption. Two cases stand out in this context: bioinformatics applications in the scientific research arena, and risk and compliance applications in industry. In this paper, we focus on the latter.

Semantics relate to the meaning and use of data. So naturally, characteristics of a domain plays an important role in determining whether a Semantic Web technology is a natural fit for applications and can help address challenges in that domain. Today, ontology is at the heart of any significant Semantic Web technology and solution. Hence a key feature that would make a semantic technology appropriate is the ability to create and manage a large populated ontology for addressing the application requirements. An ontology populated with the domain knowledge provides a critical differentiator for Semantic Web technology in solving problems where other technologies would significantly suffer due to the lack of it. We take the position that semantic technologies that utilize ontology and core technical capabilities such as knowledge representation, entity identification, disambiguation and reasoning that exploit relationships, is of primarily commercial interest for now, whether on not they already use contemporary Semantic Web language standards such as OWL. Albeit the use of standards, especially RDF/RDFS, is highly desirable for interoperability, reuse, commercialization and market adoption reasons[3].

While the technical considerations make a technology appropriate to solving a problem, no less important is the non-technical, business issue of market pull or readiness of the businesses to accept new technologies. Unique market circumstances create new opportunities and raise the needs for new applications, which can often break the lethargy or resistance in adopting new technologies and solutions. Again in this case, the risk and compliance market

[1] W3C Semantic Web Activity http://www.w3.org/2001/sw/

[2] Creative Commons License RDF validator: http://validator.creativecommons.org/

[3] We term the semantic technology that also uses contemporary Semantic Web languages and standards, namely RDF and OWL, as Semantic Web technology. However, for this paper, we will not seek to make significant distinction between the two.

has the external impetus to look for new solutions that traditional technologies do not adequately solve.

This paper deals with the discussions on the needs in the risk and compliance market that uniquely positions the Semantic Web technology as the most appropriate technology, and further gives insights into some of the key technical requirements for which a semantic approach is ideally suited. In brief, this paper seeks to explore or answer the following questions: What are the requirements and characteristics of the risk and compliance market that makes it well suited for Semantic Web technology? What are the technical capabilities of a suitable Semantic Web technology for addressing demanding and unique requirements for applications in this market?

Section 2 characterizes the market in terms of applications. Section 3 focuses on unique requirements for analytics, especially in finding links between heterogeneous data and ontological knowledge. Section 4 discusses key reasons why Semantic Web technology is an excellent fit to address the requirements. Section 5 discusses technical capabilities of a commercial Semantic Web technology. Section 6 briefly describes one application case study.

2. NEW OPPORTUNITIES AND CHALLENGES IN RISK MANAGEMENT AND COMPLIANCE MARKET

There is an unprecedented interest in the risk and compliance applications, especially in financial and government sectors. Two events and circumstances indeed shaped the corresponding market:

(a) September 11, 2001 and ensuing focus on intelligence analysis and fighting terrorism, leading to the USA Patriot Act of 2001.

(b) Corporate scandals and the need for better financial controls and corporate governance resulting from increased regulatory vigor, leading to the *Patriot Act of Finance*, the Sarbanes Oxley Act of 2002.

Correspondingly, many direct and indirect applications have appeared or are being developed. Here are just a few:

<u>Identity and Risk Management</u>: Know Your Customer (KYC) or Customer Identification Program (CIP) applications which the financial organizations are required to perform as part of the Patriot Act section 326 provisions and corresponding European Union regulations

<u>Security Screening</u>: Airport Security Screening or Passenger Threat Assessment applications, to determine if a passenger is directly or indirectly related to any known black listed entities (counties, organizations, people, etc)

and other security and prevention applications needed to support homeland security [Avant et al 2002]

Regulatory Compliance: applications supporting governance and accounting, linking data and processes to comply with the provisions of the Sarbanes Oxley Act [Ruh 2004]

Fraud Prevention: Anti-Money Laundering (AML) application[4], for example, to help avoid risks associated with doing business with customers (current or potential) who might have links with black listed entities, as required by the Patriot Act as well as European Union's Money Laundering Directive

Financial Crimes Enforcement: such as enforcement of the provisions of Section 314 of the Patriot Act[5] requiring identification and collection of evidences related to hawala operation involving a sanctioned country, arms trafficking, alien smuggling resulting in fatalities, international criminal network involved in identity theft and wire fraud, and others

Background checks and clearance: for obtaining or renewing security clearances for government jobs, the agencies need to perform substantial background checks on existing and potential employees

Authorized Information Access: for compliances with regulations such as Executive Order on Access to Classified Information[6]) or "need to know" support ensuring that employees access only that information which are necessary to perform their assignments [Aleman et al 2005]

The factors that make the business opportunity for developing risk and compliance applications for financial and government sectors more attractive include the following:

- the institutions are largely unprepared and ill-equipped to deal with the spate of significant new regulations resulting from unexpected circumstances
- the time available to implement a compliance process is in months rather than years, that the risk of non-compliance results in unacceptably high risk (i.e., the solution is *an aspirin, not a vitamin*), and
- the amount of effort involved or time for performing the required compliance activity practically argues for an automated process rather than a manual process.

A risk and compliance process usually span a number of information and knowledge driven activities, including

- identifying reliable information,

---

[4] http://www.semagix.com/solutions_circas.html

[5] FinCEN's 314(a) Fact Sheet, Financial Crimes Enforcement Network, http://www.fincen.gov

[6] Executive Order on Access to Classified Information http://www.fas.org/sgp/clinton/eo12968.html

- converting it to a usable form,
- comprehensively analyzing it with respects to mandated and optional objectives,
- identifying relevant actionable information, and
- promptly providing information or action directives to those who need it most, document the results and following up with actions varying from notification, prevention to enforcement.

The problem facing users of risk and threat assessment solutions is that, the information that powers such systems is derived from multiple sources-- typically have to be sourced both sourced internally and externally, and is heterogeneous. The challenge becomes how to drive information relevance in much focused domains and then score that information, making it available consistently and in a timely manner.

Information is the key to practically all risk and compliance processes. Following observations outline the complexity of any information processing support for vast majority of applications we outlined above.

- The type of information spans data in its raw form or factual information, as well as domain knowledge and policy descriptions
- Information (data and knowledge) is distributed with the enterprise and information providers, as well as across the open Web. Furthermore, there are different levels of autonomy and control over information sources, varying from internal and proprietary, licensed and subscribed, government and non-government agency supplied as well as open unrestricted information sources.
- Information is heterogeneous in format (unstructured in different file and application specific formats, semi-structured including static and dynamic web pages, and structured including traditional databases)
- Information is often of poor quality and of varying reliability ("Data is difficult to access, and even when it is accessible tends to be dirty or downright inaccurate" [Butler 2005])
- Information is static, time sensitive and dynamic (e.g., news and reports are made available any time), knowledge changes (a new hawala scheme is identified, an organization is added to a black list, policy is updated).

Traditional search techniques do a poor job in supporting risk and compliance applications because of the lack of context, often returning irrelevant or too much information, and without proper ranking or prioritizing. To address this problem space, there is a need to move up the continuum from pure data, to traditional search, to intelligent search utilizing metadata, semantic categorization and finally custom ontologies.

3. BEYOND SEARCH -- TO ANALYTICS VIA INTEGRATION

It is important to note that performing a good search (even when dealing with all varieties of information above) is not sufficient, and that analytical capabilities are critical for these class of applications, without which humans would be inundated with lots of irrelevant information and would not be able to implement policy or regulation uniformly across the organization. Thus the organizations who started with providing their employees the ability to crawl data sources or launch search queries against multiple web sites and data sources have quickly realized that they cannot scale effectively. However, to effectively carry out analysis, we need to integrate heterogeneous multi-source information. In other words, applications encompass search, integration as well as analytics in a highly complex information system. In this context, risk and compliance applications impose much more demanding requirements compared to a vast majority of traditional IT applications that address well defined problems in well controlled environments with limited types of information. Thus, compared to mainstream applications such as inventory management, customer relationship management, order fulfillment and human resource management, risk and compliance applications share more characteristics with the new breed of applications such as business intelligence and knowledge discovery, while not being limited to already integrated (e.g., warehoused) internal and structured data sources.

Analysis of heterogeneous information in these applications involves linking information conveyed by separate independent sources. Furthermore, identifying what is an interesting, important or material link (relationship) is the key. For example, EU Third Money Laundering Directive requires that banks formally introduce a "risk sensitive" approach to customer identification. Also necessary is the ability to focus on critical insight and drill down to arbitrary levels of detail, and translate the insight or discovery into action.

Beyond these unique challenges, these applications do share requirements posed on other enterprise applications, such as ability to do process request in batch mode, scale to millions of documents and gigabytes or terabytes of data, maintain and provide provenance of information, support the workflow that can be adapted to suite organizational structures as well as changing regulatory directives, recording in the process each critical activity for auditing, and so on.

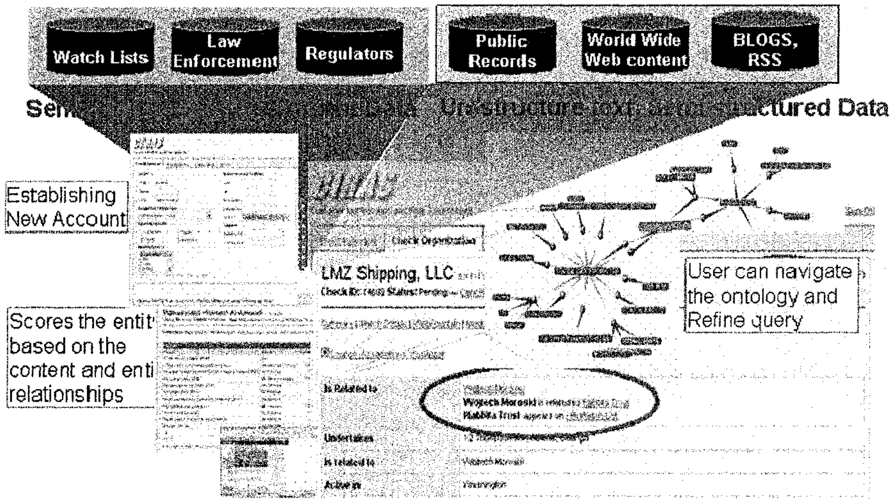


Figure 1: Schematic of a Fraud Prevention application showing heterogeneous information and ontology driven analysis

4. THE APPEAL OF SEMANTICS, ONTOLOGIES AND SEMANTIC WEB TECHNOLOGIES

There are several conceptual and fundamental reasons why semantics, ontologies and the Semantic Web technology are quite possibly the best match for risk and compliance applications.

Relationships are at the heart of semantics. For example, RDF, which is considered as a baseline (representation language) for the Semantic Web, treats relationships as the first class object, which more traditional data representation (e.g., relational model or XML) does not. For a risk and compliance application, linking relevant entities and information is at the heart of required analysis. So the Semantic Web technology is well suited to support this requirement.

It is well known that a syntactic approach grossly fails to make heterogeneous information useful, and that syntactic metadata adds very limited value. A semantic approach is necessary to integrate heterogeneous information. It is very difficulty to directly analyzing heterogeneous information, so a more appealing approach is to create semantic metadata which describes information at a more uniform level of abstraction, is domain specific and contextually relevant (as supported by an ontology).

At a more fundamental level, identification or extraction of semantic metadata require two core capabilities: entity recognition/identification (recognizing an object of interest, such as name, organization, event, etc.) and

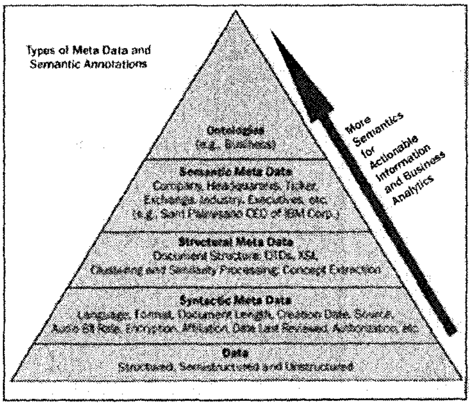


Figure 2: Metadata Semantics (From Syntax to Semantics) [Sheth 2003]

semantic disambiguation (are two objects with the same syntax -- name or description -- also same in the real world or are they different? If the ontology knows of two Bob Smiths, who does the mention of "Bob Smith" in a text refer to? Is Tiger Woods mentioned in the marketing context or the golf context?). Disambiguation is also a critical capability necessary to help build large populated ontologies, as well as deal with dirty data or conflicting information. These capabilities are important building blocks of any Semantic Web technology for enterprises.

Ontology is at the heart of the Semantic Web approach. Ontologies populated with domain knowledge become the key differentiator and enabler for core capabilities that are made possible by what we call explicit semantics (based on formal languages and domain knowledge), compared to implicit semantics (often based on statistical and learning techniques). Ontology and semantic metadata also play a critical role in defining and using context. Context enables scoring and ranking of the most important information and the analysis in help building a 360 degree perspective on an object of interest.

5. TECHNICAL CAPABILITIES OF AN ENTERPRISE SEMANTIC WEB TECHNOLOGY/PLATFORM

We first briefly discuss semantic capabilities, followed by the enterprise software capabilities, both of which are a necessary part of an enterprise grade semantic technology.

5.1 Semantic Capabilities

Earlier we described an excellent match between a semantic approach with the requirements of risk and compliance applications. The corresponding application development lifecycle is depicted in Figure 3. A Semantic Web technology needs to support the following features and capabilities.

Design ontology schema: Ontologies necessary to support most enterprise applications are highly focused. They may be partly based on industry metadata standard but often require customization with respect to coverage and depth. We have not found a practical technology to automatically design such ontologies. So the only practical solution is to use a graphical ontology design tool.

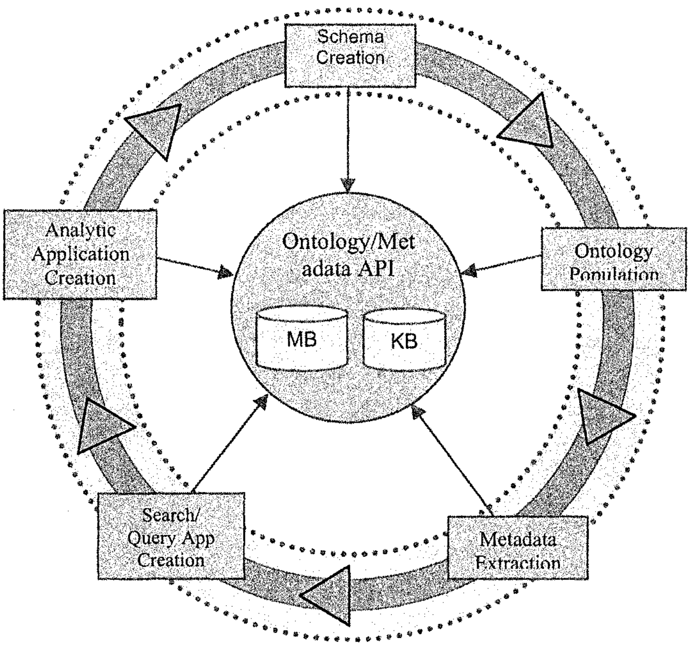


Figure 3: Semantic Application Development Lifecycle

Automatically Populate ontology with domain knowledge: Finding ontology for an enterprise application that is populated with less than a million facts (assertions, entities and relationship instances) is more an exception than a rule. Occasionally, ontology sizes approach 10 million instances. Often data (typically factual information) to populate an ontology is extracted from several trusted knowledge sources (usually data creators/aggregators to provide licensed or subscription based data, such as WorldCheck or Factiva). While knowledge sources provide structured or semi-structured information, high quality disambiguation techniques including rules that exploit provenance and trustworthiness of data, is critical for the success of automation necessary for such scales. Often it becomes necessary to use specialized disambiguation techniques and tools for matching or comparing names of persons and organization, addresses, and other types of objects. It is interesting to note, as an aside, that this approach to development of ontologies is significantly different than the social and consultative committee oriented process that is used in the development of some of the important biology ontologies and knowledgebase, such as GO and UMLS. The latter takes many years of committee effort and many million of dollars. Most ontologies for supporting industry applications need to be developed in less than three months, and are narrower in scope or coverage (focusing on an application or a class of applications). Human involvement in commercial ontology development is some times indirect - it is in the creation and curation of high quality data provided by knowledge sources, but this cost is shared across many enterprises that license or subscribe such data.

<u>Freshness of ontology</u>

Most customers require ontologies to be updated at the frequency ranging from daily to weekly. Occasionally substantial portions of an ontology may need to be refreshed and repopulated.

<u>Ontology Browsing and Visualization</u>

While software identifies actionable information or provides initial insight, it is often necessary for humans to browse, validate and drill down information. Furthermore it is necessary to be able to easily see original source of information or raw data, as well as traverse related interlinked data and information.

<u>Semantic Metadata Extraction from Heterogeneous Information</u>

A broad variety of heterogeneous information as discussed earlier needs to be processed to extract the semantics with the help of an ontology, resulting in semantic annotation or semantic metadata extraction. Although third party tools are available to deal with proprietary file formats and text conversion, processing unstructured data presents the most challenge. Automatic classifications of unstructured data can improve search, but otherwise have little value in analyzing information. Our experience shows that statistical and learning techniques (including clustering, SVM) are of little value by themselves, and that populated ontologies (i.e., a knowledge based approach) provide the most important basis for entity identification/recognition to extract metadata that is of particular interest to the application. Again disambiguation techniques are also important here. Availability of schema or discernable structure in structured and semi-structured data make is somewhat easier to ingest and process them for metadata extraction.

<u>Semantic Query and Rule Processing</u>

To enable analytical processing, the Semantic Web technology needs to provide comprehensive API for manipulating metadata and ontology, supporting the ability to efficiently process graph oriented information (including graph traversal and path computation). A number of research systems exist for RDF data storage and query processing, which are also likely to be part of future commercial systems (given numerous various of RDF query languages, completion and recommendation of SPARQL by W3C will accelerate commercial support). Support of complex queries involving both metadata (of heterogeneous data) and ontologies—for example, find the stories on the competitors of Intel (where metadata indicates the company that a story is about, and competition relationship is available from the ontology)—is especially important. For performance reasons, Semagix Freedom (a semantic application development platform from Semagix [Sheth et al 2002]) also uses main memory query processing techniques, as traditional database query processing does not given adequate performance.

Reasoning and Analytical Processing

Two types of information processing are possible. If a formal representation such as description logic (e.g., OWL) is used, inferencing is possible. However, in the context of risk and compliance application, the predominant requirement for analytic processing translates to graph oriented or link traversal type of processing. Inferencing based on subsumption does not help. Furthermore, analytical processing can be of the investigative type or the discovery type. Majority of analytical processing today is investigative type, and involves specification of rules identifying links, relationships or patterns of interest and importance. Efficient graph traversal and rules processing is thus an important capability needed for today's advanced risk and compliance applications. Discovery type of processing is an important area of research [Anyanwu 2003] and its support is in its infancy in the current commercial Semantic Web technology.

5.2 Enterprise Software capabilities

Semantic Web technology provides the cutting edge capability needed for risk and compliance applications, and in fact offers critical differentiation. At the same time, it is necessary to support capabilities enterprise users require and demand. Among the capabilities needed include both generic capabilities as well as vertical market specific capabilities. Examples of generic capabilities include:

- flexible, intuitive and highly functional user interface,
- user management (users have different levels of authority, some information is only visible to supervisors and some tasks can only be performed by authorized personnel),

- batch processing (that ability to submit a number of application queries that are then broken up into a series of tasks including semantic query),
- session management (many tasks can be interrupted and the ability to resume at a later stage is important),
- scalability (in many respects, include ability to ingest very large amount of data and large files),
- robustness with round the clock processing support (hence minimal maintenance window, and a need for redundancy),
- system monitoring, reporting, single sign-on and security, and
- use of enterprise class platforms and development methodologies.

Additionally, enterprise software also needs to deal with technology, domain and market specific capabilities. Examples of technology specific capabilities are the support for W3C standards such as RDF and OWL for Semantic Web and WSDL and SOAP for service oriented architecture.

Every risk management project and every enterprise has its own definition of what it perceives as risk. Their perception of risk is best conveyed by means of business rules that can define different scenarios and the corresponding score/action if that scenario is true or false. This calls for a comprehensive risk scoring framework that supports risk specifications which often vary drastically across projects. Also necessary is an ability to support flexible workflows that respect organizational constraints and domain or application specific routing of work, including the ability to deal with escalation of cases and exceptions.

Additional examples of domain and market specific capabilities include name normalization, identity verification, etc. One important capability is that of accessing multiple external systems, often providing the same type of service. For example, ID verification and address verification may be performed by one or more external solution providers. If there are more than one ID Verification Services, the system also needs to perform on-the-fly disambiguation of all the query results.

6. CASE STUDY: CIRAS

Regulations like European Money Laundering Directive and Section 326 of the USA Patriot Act require that financial institutions implement an Anti-Money Laundering (AML) solution. When it comes to money laundering, prevention is definitely better than cure. Detecting it after the event is simply too late, and the consequences can be devastating – both financially and in terms of an enterprise's reputation. While meeting compliance requirements and eliminating money laundering, a comprehensive Know Your Customer

(KYC) process is increasingly valuable [Levy 2004], both in terms of push (as governments introduce increasingly stringent regulations demanding that financial institutions know their customers) and pull (since a richer understanding of an organization's customers creates enormous business opportunities – in terms of modeling new services to the market at large). The Semagix Customer Identification and Risk Assessment Solution (CIRAS) is an example of a comprehensive semantic technology based solution that enables an organization to quickly and easily identify high risk customers, provides comprehensive analysis tools to perform end-to-end knowledge discovery, vastly reducing the compliance risk to the organization.

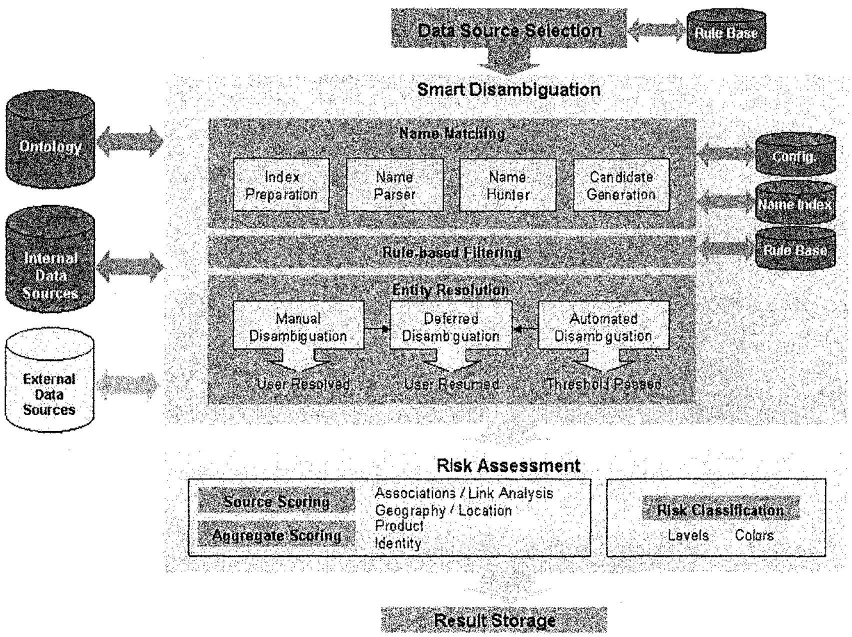


Figure 4: CIRAS KYC process

In order to implement a KYC process successfully, organizations have to bring together vast amounts of very disparate data about their customers. More importantly, though, they need to be able to make sense of all that data and content. Figure 4 shows a schematic of the KYC process engine CIRAS supports.

CIRAS is implemented using the Semagix Freedom semantic application development platform [Sheth et al 2002] with origins in the research at the LSDIS lab of the University of Georgia. Freedom provides the ontology-driven application development platform. Both the semantic and enterprise software capabilities are extensively utilized in realizing the CIRAS KYC process. The following provide additional context on some of these capabilities:

- Ontology development including knowledge base population and automatic refresh from multiple trusted knowledge sources. This involves use of external sources (as required by the organization) such as WorldCheck, OFAC, and Factiva. These contain names, organizations, aliases, watch-list membership, associations with other individuals, and other information. While ingesting relevant data using extraction agent software to populate the risk ontology, the underlying semantic technology needs to support a comprehensive disambiguation capability, including rule-based techniques. Extraction agents also run periodically as scheduled or on demand to update the ontology based on updates to the knowledge sources.
- Process/analyze wide variety of heterogeneous, multi-source information, including unstructured information (text documents, reports/documents in 150 formats), semi-structured information (Websites, emails), and structured information (databases and XML feeds) for metadata extraction as well as adapters to query data sources on-demand
- Integration with external and third party services such as ID verification (to find if a named entity is that of a recognized real world entity) and custom name matchers using flexible adapters
- Semantic processing capabilities including: entity recognition, entity resolution/disambiguation (covering scenarios such as automated disambiguation (threshold resolved), manual disambiguation (user resolved), deferred disambiguation (user resumed); risk assessment scoring using source scoring (e.g., based on geographical location), aggregate scoring (link analysis and associations), and risk classification using custom rules, provenance, etc.

In summary, unique market conditions, importance of linking and analyzing heterogeneous data, and other advanced technical requirements related to the risk and compliance applications have provided an excellent show case for the emerging Semantic Web technology. Such experiences in building semantic applications using enterprise class software is sure to lead to further successes in many other markets.

Acknowledgements

Special thanks to colleagues at Semagix, specifically Tom Golding for pointers to information on the risk and compliance market, Vernon Lun for observations on industry and application requirements, and Yash Warke for product information and the figures.

References

[Aleman et al 2005] B. Aleman-Meza, et al, An Ontological Approach to the Document Access Problem of Insider Threat, Proceedings of the IEEE Intl. Conference on Intelligence and Security Informatics (ISI-2005), May 19-20, 2005 (Springer LCNS v3495, Apr 2005, Pages 486 – 491). OpenURL

[Anyanwu 2003] Kemafor Anyanwu and Amit Sheth, "The ρ Operator: Discovering and Ranking Associations on the Semantic Web," The Twelfth International World Wide Web Conference, Budapest, Hungary, May 2003, pp. 690-699. http://lsdis.cs.uga.edu/library/download/AS03-WWW.pdf

[Avant et al 2002] David Avant, et al., Semantic technology applications for homeland security. In *Proceedings of the Eleventh international Conference on information and Knowledge Management*, McLean, Virginia, USA, November 04 - 09, 2002, ACM Press, pp. 611-613. http://doi.acm.org/10.1145/584792.584893

[Butler 2004] Martin Butler, Uncertain Compliance, Information Economics Journal, September 2004, pp. 7-9, http://www.stercomm.co.uk/download/sterling_commerce_IEJ_SEP04.pdf

[Miller 2005] Eric Miller, The Semantic Web is Here, keynote at the Semantic Technology Conference, San Francisco, CA, March 8, 2005 http://www.w3.org/2005/TALKS/0308-SEMWEB-EM.

[Lee 2005] Yvonne Lee, Apps Make Semantic Web a Reality, SDTimes, April 1, 2005 http://www.sdtimes.com/article/story-20050401-05.html

[Levy 2004] Larry Levy, Semagix AML Solution - Value of a risk approach, Financial Solutions International - Spring 2004. http://wwwt.semagix.com/documents/FIJArticle_FSI9.pdf

[Ruh 2004] William Ruh, The Web of Meaning: The Business Value of the Semantic Web, talk at the W3C 10th Anniversary (W3C10) Symposium, December 1, 2004 http://www.w3.org/2004/Talks/w3c10-WebOfMeaning/Originals/Ruh.ppt

[Sheth et al 2002] Amit Sheth, Clemens Bertram, David Avant, Brian Hammond, Krys Kochut, Yashodhan Warke, Managing Semantic Content for the Web, IEEE Internet Computing, July/August 2002, pp. 80-87

[Sheth 2003] Amit Sheth, Semantic Metadata for Next-Gen Enterprise Information Integration, DM Review, July 2003.

[Sheth 2005a] Amit Sheth, From Semantic Search & Integration to Analytics, in Semantic Interoperability and Integration, Dagstuhl Seminar Proceedings 04391, Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany, January 1, 2005.

[Sheth 2005b] Amit Sheth, Semantic Technology in the Real World: Experiences and Lessons based on Enterprise Semantic Applications, http://www.semagix.com/documents/AmitSheth-SemTechConf-Mar082005.pdf

Brief Bio:

Dr. Amit P. Sheth is a co-founder and CTO of Semagix, Inc., a professor of Computer Science and the director of the Large Scale Distributed Information Systems (LSDIS) lab at the University of Georgia. He is the Editor in Chief of the International Journal on Semantic Web and Information Systems. His research has led to two companies, three major commercial products, two patents numerous commercial applications, and over 200 (co-)authored publications. He has (co-)chaired or organized 20 international conferences and workshops, has served on over 100 program committees and serves on five journal editorial boards. He received BE from BITS, Pilani, India, and MS and PhD from the Ohio State University, USA. http://lsdis.cs.uga.edu/~amit.

PART 2
CONTRIBUTIONS

PRACTICAL DESIGN OF BUSINESS ENTERPRISE ONTOLOGIES

Tatiana Gavrilova; David Laird
Intelligent Computer Technologies Dept. Saint-Petersburg State Polytechnic University[1]; School of Information Science, University of Pittsburgh[2]

Abstract: This paper presents one approach for developing enterprise ontologies. The underlying research framework is pursuing a methodology that will aid the process of knowledge structuring and practical ontology design, with emphasis on visual techniques. For illustration of the proposed technique, the development of a practical ontology of information technology skills for a human resources knowledge management system is described.

Key words: Ontology, Visual Knowledge Engineering, Knowledge Acquisition, Knowledge Management

1. INTRODUCTION

Top managers and IT analysts are continually challenged by the need to analyze massive volumes and varieties of multilingual and multimedia data. This situation is not limited to e-business, but is seen in nearly all companies and institutions. Challenges have fueled opportunities for analytic tool developers, educators, and business process owners that support analytic communities in the management of knowledge, information and data sources. Company staff and employees require support and guidelines for knowledge sharing about information analysis, theories, methodologies and tools. Knowledge management (KM) is one of the powerful approaches to solve these problems.

The idea of using visual structuring of information to improve the quality of user learning and understanding is not new. Concept mapping has been used for more than twenty years[1,2,3] in system design and development for providing structures and mental models to support the knowledge sharing process. As such, the visual representation of general corporate business concepts facilitates and supports company personnel understanding of both substantive and syntactic knowledge. An analyst serves as a knowledge engineer by making the skeleton of the company's data and knowledge visible, and showing the domain's conceptual structure.

At the present time, this structure is called an ontology. However, ontology-based approaches to business are relatively new and fertile research areas. They originated in the area of knowledge engineering[4,5], then evolved into ontology engineering[6,7].

The discipline of Knowledge Engineering traditionally emphasized and rapidly developed a range of techniques and tools including knowledge acquisition, conceptual structuring and representation models[8,9]. These developments have underpinned an emerging methodology that can bridge the gap between the ability of the human brain to structure and store knowledge, and the knowledge engineers' ability to model this process. But for practitioners, knowledge engineering is still a rather new, eclectic domain that draws upon a wide range of areas, including cognitive science, etc. Accordingly, knowledge engineering has been, and still is, in danger of fragmentation, incoherence and superficiality.

Since 2000, a major interest of researchers has focused on building customized tools that aid in the process of knowledge capture and structuring. This new generation of tools – such as Protégé, OntoEdit, and OilEd - is concerned with visual knowledge mapping that facilitates knowledge sharing and reuse[10,11,12]. The problem of iconic representation has been partially solved by developing knowledge repositories and ontology servers where reusable static domain knowledge is stored. Ontolingua, Ontobroker and many others are examples of such projects[13,14].

The usage of ontologies has special value in companies where specialists reuse domain ontologies in order to support the business protocols that are grounded in the domain's problem-solving methodology. Therefore, the basic idea is to allow experts to model both domain and problem-solving knowledge using the same visual language. Knowledge entities that represent static knowledge of the domain are stored in hierarchical order in the knowledge repository and can be reused by others. At the same time, those knowledge entities can also be reused in description of the properties or methodological approach as applied in the context of another related knowledge entity. The involved concept map modeling language is based upon a class-based object-oriented language that supports the classification and parameterization of knowledge entities.

This paper proposes a practical approach to business ontology design. The underlying research is pursuing usage of visual iconic representation and diagrammatic structures, with emphasis on visual design. For clearer understanding of the methodology, the process of developing a practical ontology of information technology knowledge and skills is described. In the remainder of the paper, we will describe some theoretical issues regarding ontological engineering and present our proposed methodology for ontology design. Moreover, we will describe our detailed practical example using the proposed methodology. In conclusion, we provide insight through discussion of current and possible future work.

2. USING ONTOLOGICAL ENGINEERING FOR BUSINESS APPLICATIONS

We start the discussion of theoretical issues of ontological engineering by developing a definition of ontology from literature within the field.

2.1 Ontology Definition

Ontology is a set of distinctions we make in understanding and viewing the world. There are numerous well-known definitions of this seminal term[15,16,17], that may be generalized by such:

"Ontology is a hierarchically structured vocabulary describing a domain that can be used as a skeletal foundation for a knowledge base".

This definition clarifies the ontological approach to knowledge structuring while providing sufficient freedom for open-ended, creative thinking. For example, ontological engineering can provide a clear representation of a company's structure, human resources, physical assets, and products, and their inter-relationships. Many researchers and practitioners argue about the distinctions between the ontology and the user's conceptual model. We believe that the ontology corresponds to the analyst's view of the conceptual model, but is not the de facto model.

Ontology as a useful structuring tool may greatly enrich the business modeling process, providing users of KM-systems an organizing axis to help them mentally mark their vision of the domain knowledge.

2.2 Creating Ontologies for Business Use

Ontology creation faces the knowledge acquisition bottleneck problem. The ontology developer frequently encounters the additional problem of not utilizing sufficiently tested and generalized methodologies, which would recommend what activities to perform and at what stage of the ontology development process. An example of this can be seen when each development team generally follows its own set of principles, design criteria, and steps in the ontology development process. The lack of structured guidelines and methods hinders the development of shared and consensual ontologies within and between the teams. Moreover, it makes the extension of a given ontology by others and its reuse in other ontologies and final applications difficult[18].

Several effective domain-independent methodological approaches have been reported for building ontologies[6,7,19]. What these approaches have in common is that they consistently begin with identification of the purpose of the ontology, and the need for acquisition of the domain's knowledge. However, having acquired a significant amount of knowledge, major researchers propose a formal language expressing the idea as a set of intermediate representations and then generating the ontology using translators. These representations bridge the gap between how people see a domain and the languages in which ontologies are formalized. The conceptual models are implicit in the implementation codes. A re-engineering process is usually required to make the conceptual models explicit. Ontological commitments and design criteria are implicit in the ontology code.

Figure 1 presents our vision of the mainstream state-of-the-art categorization in ontological engineering[20,21,22] and may help the knowledge analyst to figure out what type of ontology he/she really needs. We use Mindmanager™ as it proved to be a powerful visual tool.

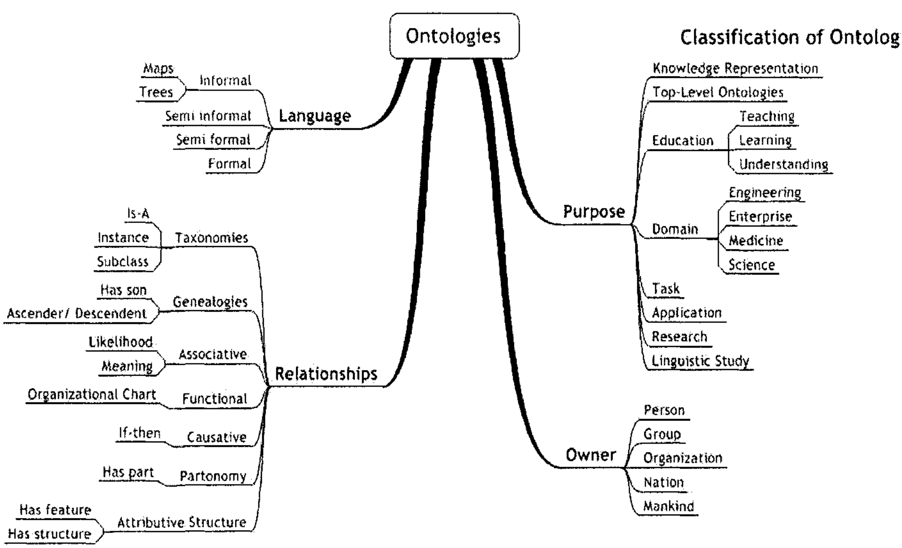


Figure 1. Ontology classification

Frequently, it is impossible to express company business information in a single ontology. Accordingly, company knowledge storage consists of a set of related ontologies. However, some problems may occur when moving from one ontological space to another that could be solved by constructing meta-ontologies that may help to resolve these problems.

We can propose different types of ontologies that can support business applications:

- Company organizational structure
- Main concepts ontology (products, services, customers, skills, etc.),
- Historical ontology (genealogy of owners, customers, products, services, etc.),
- Partonomy of the company knowledge
- Taxonomy (methods, techniques, technologies, business-processes, skills, etc.)

The concrete set of ontologies depends on personal vision, business application and awareness level of the system's analysts and users. Generalizing our experience in developing different business and teaching ontologies in the field of consulting, business modeling and information technologies[23,24,25,26], we propose a four-step algorithm that may be helpful for visual ontology design. We try to develop the ideas of Uschold and King's skeletal methodology[22] putting stress on details of ontology capture, where visual representation works as a powerful mind tool[2] in the structuring process. Visual form influences both analyzing and synthesizing procedures in ontology development process. That is why we believe that the "beauty" of the ontology plays an important role in understanding of the knowledge.

3. ONTOLOGY CREATING

While in major works the emphasis is put on ontology specification, we would like to elucidate the essentials.of ontology capture[22], not coding.

3.1 Four-Step Algorithm

Stepl. Goals, strategy and boundary identification: The first step in ontology development should be to identify the purpose of the ontology and the needs for the domain knowledge acquisition. It is important to be clear about why the ontology is being built and what its intended uses are[22]. We also need to define the scope or "boundaries" of the ontology, before compiling a glossary. It is also important to elucidate the type of ontology according to Figure 1 classification, such as taxonomy, partonomy, and genealogy. That effort is done at this step, as it affects the next stages of the design.

Step2. Glossary development or meta-concept identification: This time consuming step is devoted to gathering all the information relevant to the described domain. The main goal of this step is selecting and verbalizing all of the essential objects and concepts in the domain. A battery of knowledge elicitation techniques may be used – from interviews to free association word lists.

Step3. Laddering, including categorization and specification: Having all the essential objects and concepts of the domain in hand, the next step is to define the main levels of abstraction. Consequently, the high level hierarchies among the concepts should be revealed and the hierarchy should be represented visually on the defined levels. This could be done via a top-down strategy by trying to break the high level concept from the root of the previously built hierarchy, by detailing and specification of instance concepts. Revealing a structured hierarchy is one of the main goals at this stage. Another way is generalization via bottom-up structuring strategy. Associating similar concepts to create meta-concepts from leaves of the aforementioned hierarchy could do this. The main difficulty is forming categories by creating high level concepts and/or breaking them into a set of detailed ones where it is needed.

Step4. Refinement: The final step is devoted to updating the visual structure by excluding any excessiveness, synonymy, and contradictions. As mentioned before, the main goal of the final step is try to create a beautiful ontology. The ideas of "beatification" are well known in basic studies beginning from the search for beautiful formula, model or result. Beauty was always a very strong criterion of scientific truth. We believe that harmony and clarity are what make an ontology beautiful.

3.2 Harmony

To achieve harmony, we attempt to follow Gestalt (good form) principles by M. Wertheimer[27]:

- Law of Pragnanz: organization of any structure in nature or cognition will be as good (regular, complete, balanced, or symmetrical) as the prevailing conditions allow (law of good shape).

- Law of Proximity – objects or stimuli that are viewed being close together will tend to be perceived as a unit.
- Law of Similarity – things that appear to have the same attributes are usually perceived as being a whole.
- Law of Inclusiveness (W. Kohler) - there is a tendency to perceive only the larger figure, and not the smaller, when it is embedded in a larger.
- Law of Parsimony – the simplest example is best known as Ockham's razor principle (14th century): ``entities should not be multiplied unnecessarily".

3.2.1 Conceptual balance

A well-balanced ontological hierarchy equates to a strong and comprehensible representation of the domain's knowledge. Ill-balanced ontology design (at Figure 2) shows that long branches are over-detailed, while shorter ones are under-investigated. For our problem, this may create a situation where some IT skills will be described too precisely, while others will be just briefly mentioned. Ill-balanced ontology often demonstrates the low professional level of the expert and/or knowledge analyst. However, it is a challenge to formulate the idea of a well-balanced tree. Here we offer some tips to help formulate the "harmony":

- Concepts at one level should be linked with the parent concept by only one type of relationship, such as "is-a", or "has part".
- The depth of the branches should be more or less equal (±2 nodes).
- The general outlay should be symmetrical.
- Cross-links should be avoided as much as possible.

Fig.2 illustrates the balance idea.

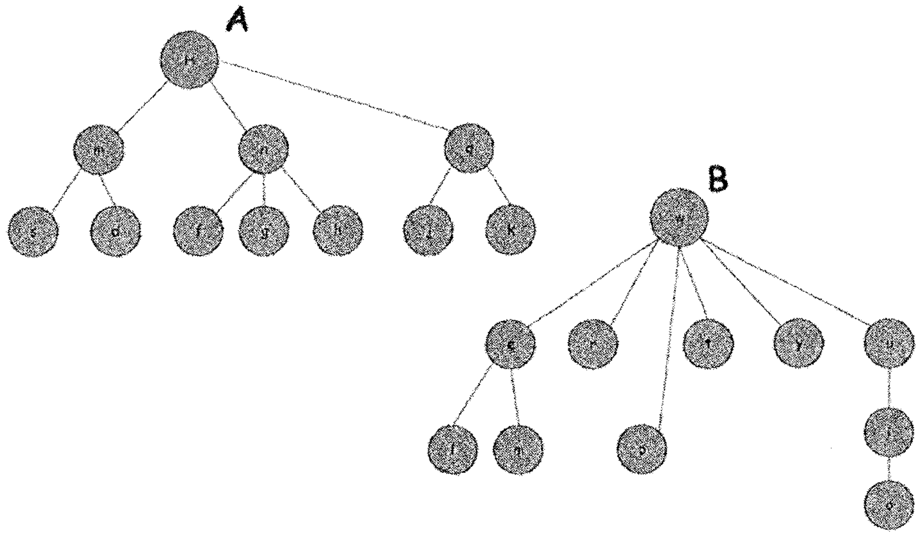


Figure 2. Well-balanced (A) and ill-balanced (B) ontologies

3.3 Clarity

In addition to the principle of harmony, it is important to pay attention to clarity when building a comprehensible ontology. Clarity may be provided through a number of concepts, and types of the relationships among the concepts.

- Minimizing the number of concepts. The maximal number of branches and the number of levels should follow Miller's magical number (7 ± 2)[28].
- Furthermore, the type of relationship should be clear and obvious if the name of the relationship is missing.

Some tips to achieve visual clarity are described later in section 4.4.

4. DEVELOPING A PRACTICAL ONTOLOGY

In this section we describe the development of an ontology of information technology skills and knowledge, following the aforementioned 4-step algorithm. We have tried to report the exact practical procedures we followed at each step by including all the visual structures.

4.1 Step 1 - Purpose and use of Ontology:

It is important to first identify the purpose and proposed usage of the ontology early in the process of its development[22]. The example ontology described throughout the remainder of this paper was developed to support a business application to address the following needs.

Situation/Problem: A company is seeking to identify the knowledge and skills of each of its employees that are relevant to the work of the company. This data will allow the company to:

- Identify the essential skills of the organization
- Develop a knowledge retention strategy to ensure that sufficient depth is present in the organization in the event of resignation, retirement, or other loss of key employees
- More effectively identify and utilize employee skillsets:
  - Allow employees to quickly find experts to address unique questions or problems.
  - Identify individuals in the company with the needed skill to work on new or expanding projects
- Develop individual and organization-wide training plans and strategy, based on the collective training needs of the enterprise.

Solution: Use a network-based intranet application that allows employees to identify their individual skills and training needs. This application will make use of an ontology of skills that span the IT industry, and allow employees to select relevant skills and knowledge from that ontological presentation of skills, which they currently possess, or have a business need to acquire. Use of the ontology in this way serves the following purposes:

- Ensures that each employee considers the entire range of IT Skills that he might possess, or that are relevant to the organization.
- Ensures that data is entered uniformly into the system by each employee, with a consistent understanding of the meaning of each skill. This consistency allows subsequent searches of the employee skills database to find all cases of a selected skill, and the organization's training to be planned for specific or broad categories of skills.
- Provides for a framework to visualize and better understand the relationships of skills that are relevant and critical to the success of the organization.

4.2 Step 2 - Glossary Development

As previously mentioned, the first step in building an ontology is collecting information in the domain and building a glossary of the terms of the domain. To build a glossary of information technology skills, we collected the terms from two different types of resources: closed-corpus material and open-corpus material.

The closed corpus materials are in the form the company's job descriptions in the field of information technology, recent correspondence and status reports from the Information Systems department, and an organizationally generated skills inventory. The open corpus materials include the table of contents and index of general information systems text references, categorizations and descriptions of computer and information science course offerings, and existing ontologies about the field of information technology. The terms and concepts from each of these sources were combined to build a single glossary.

Table 1. Sampling of a Glossary of IT Skills and Knowledge

Personal Computer Maintenance	Peripherals Maintenance	Help Desk Support
Database Administration	Programming	Application Development
Cyber Security	Encryption	Commercial Software
Super Computing	Telecommunications	Training
Project Management	Graphics	Mobile Computing
Computer Architecture	Software Development Lifecycle	Quality Assurance
Human Factors in Systems	Human Computer Interactions	Artificial Intelligence
Geographic Information Systems	Decision Support Systems	Data Mining
Information Storage and Retrieval	Programming Languages	Software Engineering
Algorithm Design	Computer Engineering	Visual Languages
Operating Systems	Document Processing	Information Processing Standards
Knowledge Representation	Legal and Ethical Issues	Expert Systems
Ontologies	Knowledge Management	Routers
Bridges	Network Switches	Computer Server Support
Virus Detection	Email Systems	Enterprise System Customization

4.3 Step 3 - Laddering: Building an Initial Mind Map Structure

In the third step, we built an initial visual structure of the glossary terms. The main goal of this step is the creation of a set of preliminary high level concepts and the categorization of the glossary terms into those concepts. A mind map can be a useful visual structure for this step. **Figure 3** presents the mind map of our initial categorization. Since the categorization in this step is preliminary, some of terms might not fit into any of the initial categorization. We should mention that the categorization in this step is done entirely manually. However, we employed the job descriptions, text glossary and table of contents, and groupings of university course offerings, which were used to build the glossary in the previous step, to build the initial categories as well. We can consider the groupings from these sources to be expert help in designing the ontology, because such groupings were accomplished to make the presented information in these sources clear and easily accessible; traits that we desire in the finished IT Skills ontology.

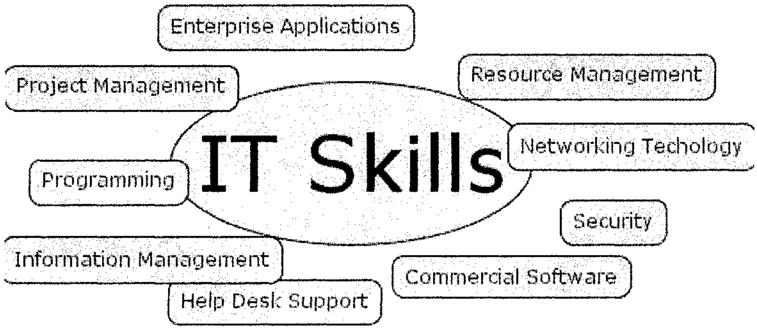


Figure 3. Trivial Categorization

Figure 4 presents the details of our initial categorization of the terms into the concept in **Figure 3**. The visual structures presented in this step illustrate the idea of how an ontology can bridge the gap between the chaos of unstructured data presented in the glossary, and be a clear means of showing mapped representations.

Later, we composed more precise concepts and hierarchies by analyzing the glossary and previously built visual structure. First we employed the top-down design strategy to create meta-concepts such as Programming, Network Support, Project Management, etc. Then using the bottom-up strategy we tried to fit the terms and concepts into the meta-concept. Moreover, we created the relationships between the concepts. A concept map is the most useful visual structure for representation of the results of this stage, since it gives the ability of defining the relationship in addition to building the hierarchy. The output of this step is a large and detailed map, which covers the domain hierarchically.

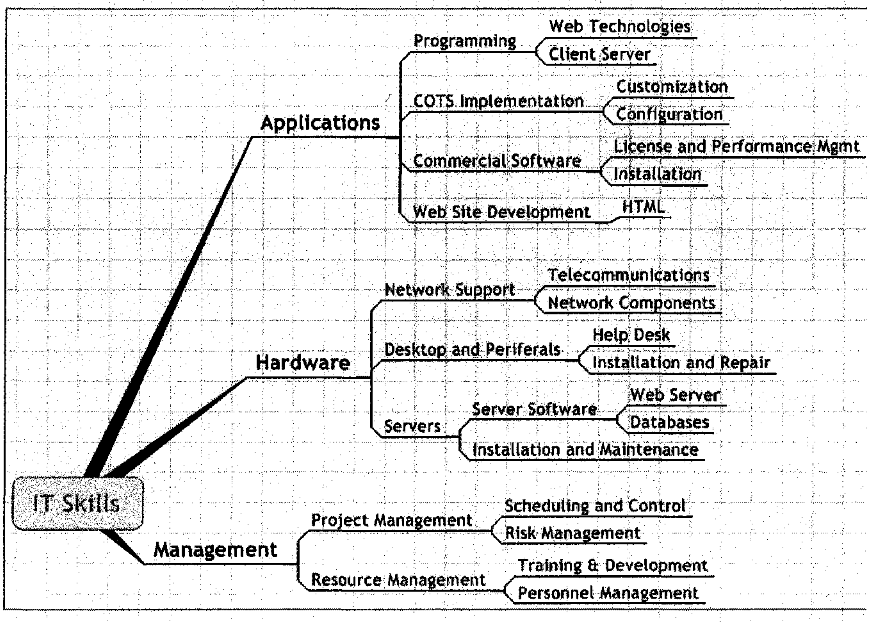


Figure 4. Details of first level categorization

Next, based on the detailed concept map, we built the general ontology that is shown in **Figure 5**, utilizing liberal relationship terms to link the concepts with detailed terms from the glossary.

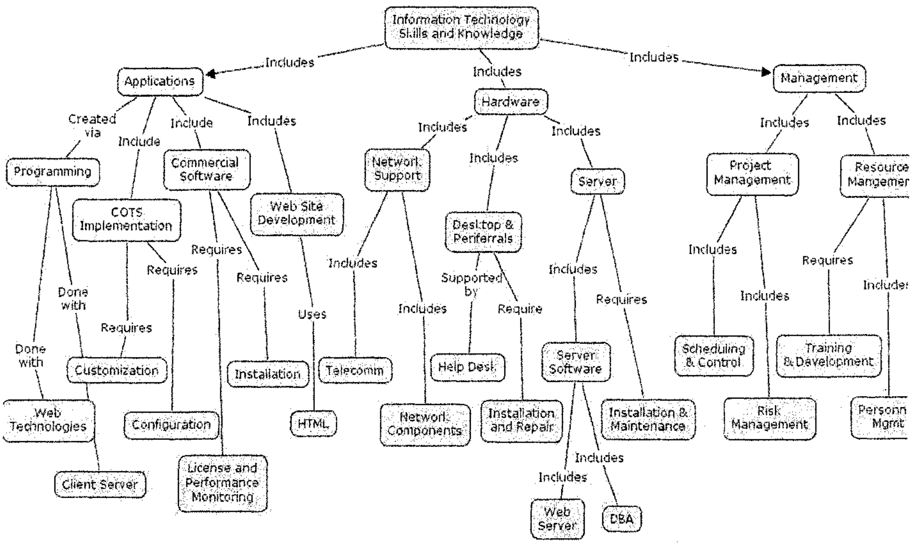


Figure 5. General ontology

4.4 Step 4: Refinement

As described in the algorithm, the final step is devoted to making the ontology beautiful. The following are some practical tips that may be taken into consideration while refining the ontology, and are illustrated in Figure 6:

1. Use different font sizes for different strata
2. Use different colors to distinguish particular subsets or branches (not very clear in the black and white printout).
3. Use a vertical layout of the tree structure/diagram.
4. If needed, use different shapes for different types of nodes.

Moreover, we re-built the general ontology while taking into consideration the harmony and clarity factors. Comparing **Figure 5** and **Figure 6** presents these changes. Another feature of harmony is having the same relationship in every level. Moreover, to achieve clarity, we removed all unnecessary nodes and use standard, consistent relationships to simplify understanding.

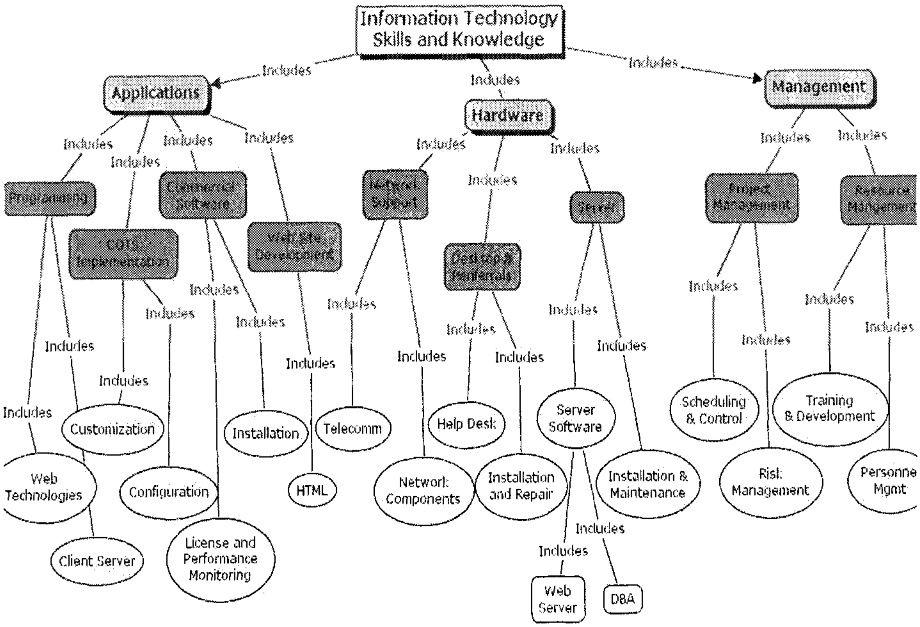


Figure 6. Harmony and clarity in the ontology

5. DISCUSSION

Our research stresses the role of knowledge structuring for developing ontologies quickly, efficiently, and effectively. At a basic level of knowledge representation, within the context of everyday heuristics, it is easier for practitioners simply to draw the ontology using conventional "pen and pencil" techniques. However, for more sophisticated knowledge representations, our proposed 4-step ontology development process is proposed.

Development and use of an ontology of IT Skills and Knowledge was illustrated in this paper to provide a concrete example of the proposed methodology. A more detailed version of the illustrated ontology will be integrated into the use of a Knowledge Management application, used to develop a map of critical skills and knowledge within a business enterprise. This awareness of the critical skills needed and possessed by the organization will allow strategies to be developed to ensure the retention and most effective use of those critical skills. Without a comprehensive ontology to frame this investigation, valid and useful results would be far more difficult to achieve.

In subsequent research, we plan to explore ways that ontology development and use can further improve visualization of business needs, and deliver additional value to the organization. Such investigations will address the reduction of overlap between business units in an organization, aligning recruiting efforts with actual business needs, development of job descriptions that accurately reflect the skills and knowledge truly needed for the success of the organization, and clearer understanding of the most critical and valued skills within the organization.

6. ACKNOWLEDGMENTS

This work is partly supported by the grant of Russian Foundation for Basic Research No.04-01-00466.

7. REFERENCES

1. J.F. Sowa (1984) Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading, Massachusetts.
2. D.H. Jonassen (1998) Designing constructivist learning environments. In Instructional design models and strategies (C.M. Reigeluth (Ed), 2nd ed., Lawrence Eribaum, Mahwah, NJ.
3. T. Conlon (1997) Towards Diversity: Advancing Knowledge-based Modelling with Knowledge Acquisition. In Proceedings of 8th International PEG Conference, Sozopol, Bulgaria., pp.379-386.
4. J.H. Boose (1990) Knowledge Acquisition Tools, Methods and Mediating Representations. In Knowledge Acquisition for Knowledge-Based Systems (Motoda, H. et al., Eds), IOS Press, Ohinsha Ltd., Tokyo, pp.123-168.
5. P. Brusilovsky and R. Rizzo (2002), Map-Based Horizontal Navigation in Educational Hypertext. In Proceedings of Hypertext, University Of Maryland, College Park, USA
6. D. Fensel (2001) Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer.
7. R. Mizogushi and J. Bourdeau (2000), Using Ontological Engineering to Overcome Common AI-ED Problems. International Journal of Artificial Intelligence in Education, volume 11, pp.1--12.
8. H. Adeli (1994) Knowledge Engineering. McGraw-Hill, New York.
9. A. Scott, J.E. Clayton & E.L. Gibson (1994) A Practical Guide to Knowledge Acquisition, Addison-Wesley.
10. Protégé, Stanford Medical Informatics. Accessed from http://protege.stanford.edu/ at December 07, 2004
11. OntoEdit, AIFB, University of Karlsruhe. Accessed from

http://www.ontoknowledge.org/tools/ontoedit.shtml at December 07, 2004

12. OilEd, S. Bechhofer and G. Ng Accessed from http://oiled.man.ac.uk/ at December 7, 2004
13. OntoBroker, Accessed from http://ontobroker.aifb.uni-karlsruhe.de/index_ob.html at December 7, 2004
14. Ontolingua, Stanford University. Accessed from http://www.ksl.stanford.edu/software/ontolingua/ at December 7, 2004
15. Neches, et al , (1991) Enabling Technology for Knowledge Sharing. AI Magazin,. Winter, pp.36- 56.
16. T. Gruber (1993) A translation approach to portable ontology specifications. Knowledge Acquisition , Vol. 5, pp.199- 220.
17. A. Gómez-Pérez, M. Fernández-López, O. Corcho (2004) Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web Springer.
18. N. Guarino & P. Giaretta (1998) Ontologies and Knowledge Bases: Towards a Terminological Clarification. In Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing, IOS Press, pp.25- 32.
19. B. Swartout, R. Patil, K. Knight & T. Rus (1997) Toward Distributed Use of Large-Scale Ontologies. In Ontological Engineering, AAAI- 97 Spring Symposium Series, pp.138-148.
20. N. Guarino, C. Welty (2000) A Formal Ontology of Properties. In R. Dieng and O. Corby (eds.), Knowledge Engineering and Knowledge Management: Methods, Models and Tools. 12th International Conference, EKAW2000. Springer Verlag: 97-112.
21. R. Jasper and M. Uschold (1999). A Framework for Understanding and Classifying Ontology Applications. In Twelfth Workshop on Knowledge Acquisition Modeling and Management KAW'99.
22. M. Uschold, M. Gruninger (1996). Ontologies: Principles Methods and Applications. In Knowledge Engineering Review, Volume 11, Number 2.
23. T.A. Gavrilova & A. Voinov (1996) Visualized Conceptual Structuring for Heterogeneous Knowledge Acquisition . In Proceedings of International Conference on Educational Multimedia and Hypermedia, EDMEDIA'96, MIT, Boston, USA, pp.258-264.
24. T. Gavrilova, A. Voinov (1998) Work in Progress: Visual Specification of Knowledge Bases, Lecture Notes in Artificial Intelligence 1416 "Tasks and Methods in Applied Artificial Intelligence", A.P.del Pobil, J.Mira, M.Ali (Eds), Springer. - pp. 717-726.
25. T.A. Gavrilova, A. Voinov, E. Vasilyeva (1999) Visual Knowledge Engineering as a Cognitive Tool, Proc. of Int. Conf. on Artificial and Natural Networks IWANN'99, Spain, Benicassim. - pp.123-128.
26. T. Gavrilova (2003) Teaching via Using Ontological Engineering, Proceedings of XI Int. Conf. "Powerful ICT for Teaching and Learning" PEG-2003, St.Petersburg. – p. 23-26.
27. M. Wertheimer (1982) Productive Thinking. Univ of Chicago Pr (Tx); Phoenix ed edition,
28. G. Miller (1956) The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. The Psychological Review, vol. 63, pp. 81-97.

RgbDF: RESOURCE GOAL AND BEHAVIOUR DESCRIPTION FRAMEWORK

Olena Kaykova, Oleksiy Khriyenko, Vagan Terziyan, Andriy Zharko
Industrial Ontologies Group, Department of Mathematical Information Technology, University of Jyvaskyla, FINLAND, e-mail: vagan@it.jyu.fi

Abstract: Agent-oriented approach has proven to be very efficient in engineering complex distributed software environments with dynamically changing conditions. The efficiency of underlying modelling framework for this domain is undoubtedly of a crucial importance. Currently, a model-driven architecture has been the most popular and developed for purposes of modelling different aspects of multi-agent systems, including behaviour of individual agents. UML is utilized as a basis for this modelling approach and variety of existing UML-based modelling tools after slight extension are reused. This paper proposes an ontology-driven approach to modelling agent behaviour as an emerging paradigm that originates from the Semantic Web wave. The proposed approach aims at modelling a proactive behaviour of (web-)resources through their representatives: software agents. In general, the presented research puts efforts into investigation of beneficial features of ontology-based agent modelling in comparison with conventional model-driven approaches.

Keywords: agents, Semantic Web, resource proactivity, goal, behaviour, ontology

1. INTRODUCTION

There is a huge amount of academic and industrial initiatives world-wide related to agent-oriented analysis. To organize these efforts, a special Co-ordination Action for Agent Based Computing, AgentLink III[1], funded by the European Commission's 6th Framework Program, was launched on 1st January, 2004 until December 2005. The AgentLink III initiative currently has registered 99 projects and 128 software products based on agent

[1] http://www.agentlink.org/

approach. Core technologies of several commercial organizations utilize different agent paradigms. For example, Whitestein Technologies[2] and Agent Oriented Software Pty Ltd[3] have provided advanced software agent technologies, products, solutions, and services for selected application domains and industries since 1999. Agent-based approach has been tried in a research of industrial automation systems domain [2, 7].

Modelling of multi-agent systems and behaviour of concrete agents in it has been one of the most significant topics in various domains. Model-driven approach to design of agent behaviours emerged a long time ago and initially was based on UML modelling [9, 11]. Later this approach was extended to a level of meta-modelling [10]. As one of the mature UML-based methodologies for modelling multi-agent systems, Agent Modelling Language can be mentioned [12]. Currently, AML is used in commercial software projects, is supported by CASE tools and in the nearest future first version of its specification will be presented to public for its further development. One of the fundamental formal theories about behaviour in multi-agent systems [13] is developed and lectured in Free University of Amsterdam[4].

All the above efforts have in details elaborated a conceptual base of agent behavioural modelling and motivated its further development. There were even attempts to elaborate a conceptual convergence of an agent layer and Web Service Architecture [8]. However, academic efforts lack concrete details concerning methodology of modelling or have just very preliminary prototype implementations as e.g. the Agent Academy project [1].

Recently, ontology-driven approach is growing as an option to Model-driven one, while having several advantages:

- Possibility of reasoning on a level of a single model and inter-model relationships and mappings, supporting meta-model level as well.
- Support of flexibility for tools (e.g. XSLT transformations) based on ontology during an evolution of the ontological model (see analysis of evolution of classes and properties and its impact on tools in [14]).
- More flexible modelling framework based on a graph [15].

DERI is among research centres that are very close to implementing really powerful prototypes of ontology-driven modelling for Web Services and Multi-Agent Systems. Significant efforts for development of agent goal-

[2] http://www.whitestein.com/
[3] http://www.agent-software.com/
[4] http://www.vu.nl/

behaviour frameworks based on WSMO standard (ontology-driven) have been conducted by a research group from DERI according to their vision of Semantic Web Fred [3].

As a possible option of implementation of agents behaviour engine are Horn-like rules. W3C standardization efforts aimed at this direction, have recently resulted into a family of standards: RuleML[5] (Rule Markup Language), SWRL[6] (a Semantic Web Rule Language Combining OWL and RuleML), FOL RuleML[7] (First-Order-Logic RuleML), SWRL FOL[8] (SWRL extension to First-Order Logic). All these standards are tightly related to previous research carried out by IBM alphaWorks Labs in development of CommonRules[9] and BRML (Business Rule Markup Language). The initiative within CommonRules was aimed at a development of a framework for specification of executable business rules by non-programmer business domain experts. The final result represented a reusable technology of business rules and rule-based intelligent agents embodied as extensible Java library. Industrial Standards, related to modelling and automation of business behaviour, are currently concentrated around BPEL4WS[10] and ebXML[11].

From the technological side, there are reliable options to be a basis for implementing frameworks for modelling behaviours in multi-agent systems: JADE-Jess-Protégé[12] and Aglets SDK[13]. In the JADE implementation several Java upper classes have been provided [4] and this promoted use of the JADE platform in implementation of tools for modelling complex agent behaviours. The JADE Framework has been extended by a BDI infrastructure within the Jadex[14] project [5] and its behavioural model was extended by Hewlett Packard Lab in their HP SmartAgent initiative [6].

The research presented in this paper aims at development of a framework for modelling ontology-driven proactive behaviour of resources using Multi-Agent Systems. This research is the part of ongoing SmartResource[15] project ("Proactive Self-Maintained Resources in Semantic Web") activities performed by Industrial Ontologies Group[16]. The grounds for ontological description of agent-based resource proactivity were prepared by previous

[5] http://www.ruleml.org/
[6] http://www.daml.org/2003/11/swrl/
[7] http://www.daml.org/2004/11/fol/folruleml
[8] http://www.daml.org/2004/11/fol/
[9] http://www.research.ibm.com/rules/commonrules-overview.html
[10] http://www-128.ibm.com/developerworks/library/specification/ws-bpel/
[11] http://www.ebxml.org/
[12] http://jade.tilab.com/doc/examples/JadeJessProtege.html
[13] http://www.trl.ibm.com/aglets/
[14] http://vsis-www.informatik.uni-hamburg.de/projects/jadex/
[15] http://www.cs.jyu.fi/ai/OntoGroup/SmartResource_details.htm
[16] http://www.cs.jyu.fi/ai/OntoGroup/index.html

related research, see e.g. [22, 23]. The aim for such "proactivity framework" is related to the industry needs to have "smart" industrial resources (devices, machines, processes, organizations, etc.), which will be able to proactively monitor, diagnose and maintain own state and condition.

2. RESOURCE GOAL AND BEHAVIOUR DESCRIPTION FRAMEWORK (RGBDF)

Autonomous systems must be automatic and, in addition, they must have a capacity to form and adapt their behaviour while operating in the environment. Thus traditional AI systems and most robots are automatic but not autonomous - they are not fully independent from the control provided by their designers. Autonomous systems are independent and are able to perform self-control. As it is argued in this paper, to do this, they must be motivated.

In Agent Environment (as well as in the real world) the base for any interaction is behaviour of each individual. Further, integration of these individual behaviours may form behaviour of Agent Alliance. In real world almost all of behaviours (actions) are goal-driven, but some of them are not. With software agents in mind we are focused just on the goal-driven behaviour. What is a goal-driven behaviour? Such behaviour means performing set of rules, which are aimed to achievement of certain goal. In return, goal is a fact which does not exist in a description of the environment, and an agent aims at appearance of the fact. As a result, we have a trio: behaviour which is driven by certain goal and which lies in performing actions following a set of behavioural rules. However, even having a rule base, which enables an agent to achieve a goal, still extra information (environmental facts) is needed. This is because each rule has to have a sufficient condition. In our case a sufficient condition is a presence of input data for action being performed. Having the sufficient condition we should take into account also a necessary condition: presence of a goal along with a certain context (set of facts of the environment) for performing the goal. Not all goals assume execution of unambiguous rule(s). Some goals can be represented by aggregation of more specific goals.

Referring to the trios that were discussed above, each agent should have initial set of those trios (regulated by initial role). These trios represent expertise and experience of an agent. As well as in real world agents can exchange their expertise (rules for execution of actions depending on the goals and direct software modules for execution of actions). Availability of a wide spectrum of the trios gives a possibility for agent to automatically divide up goals (which cannot be achieved because of lack of information) to sub goals and to create a chain of nested trios.

One more thing from a modelling paradigm that can be applied to an agent is an agent role. Agent role means aggregate of goals corresponding to a specific purpose of the agent. Individual role does not assume a fixed set of activities, the set of the goals can be different even for the same role depending on the context. Such approach to the goal and behaviour description brings a possibility for agent to be more autonomous. Through utilization of this approach agent can change its role, set of the goals corresponding to its purpose depending on a condition of the environment. In other words, an agent can change its behaviour based on a context.

Approach of RG/BDF assumes concentrating all the goals, roles descriptions and templates of behavioural rules in ontology. The templates of behavioural rules are described in a general way with a purpose to be applied to any particular agent. Such description requires utilization of a handy and flexible description schema (RG/BDFS-Lite), which will be presented later. Architecture of general Agent Platform is represented in Figure 1.

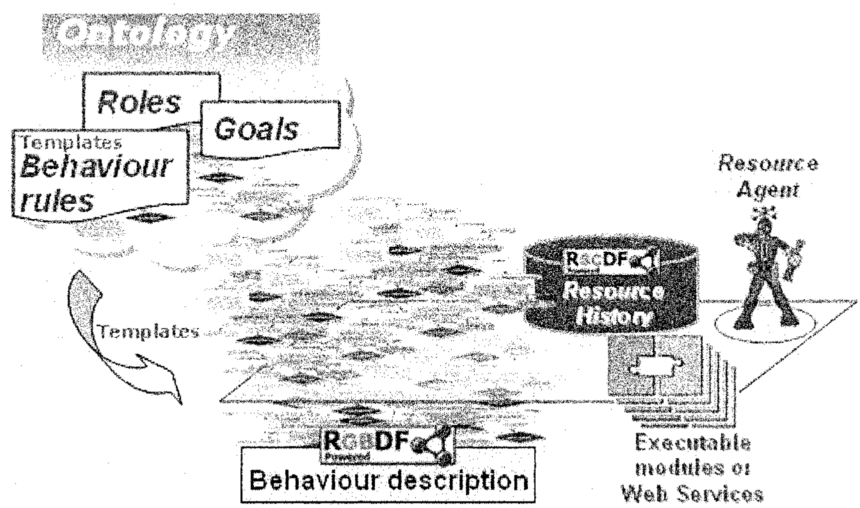


Figure 1. Architecture of the SmartResource platform

On its own platform agent has the Resource History (encoded e.g. in RscDF contextual extension of RDF [21]), where it stores all statements about resource states, conditions and actions that have been performed by the resource agent and other contextual information that can be useful (statements about the environment of the resource). Some executable modules (code) that the agent must perform also can be located there as an output of its behavioural rule chain. Otherwise the agent has to utilize external web services. Agent always has to interact with ontology server to be able to download necessary role, goal description or behavioural template whenever the agent needs it.

Behavioural template represents a rule for behaviour in RDF serialization. The template is represented by a behavioural statement *rgbdfs:Behaviour_Statement* (it will be described in the next chapter) and contains necessary condition (goal) and sufficient condition (condition of the environment) as the contexts of rule execution and a set of the executive descriptions (specification of the executable modules that should be invoked) as an output of the rule.

We can divide the process of the resource goal and behaviour annotation to several stages. The first one is a stage of goal instance definition that assumes creation (process of describing) of a statement to which an agent should strive. This goal can be specified directly by an expert or via specification of the agent goal. Based on this goal description appropriative behaviour template(s) have to be found in ontology, downloaded and transformed to the behavioural instance(s) on the resource platform. After this the needed executable modules (if they are not located at the resource platform) also can be downloaded. As a final stage of goal/behaviour annotation process expert has to specify (add/modify) the context of the behaviour. Now the platform contains behavioural rule(s) in RDF/XML serialization form that can be performed by the agent engine. This engine follows the behavioural rules till the goal is achieved.

3. RGBDFS-LITE

In continuation to the idea of Context Description Framework[17] (CDF), which was developed by Industrial Ontologies Group[18], such approach can be applied to context sensitive Resource Goal and Behaviour Description Framework (RG/BDF). RG/BDFS-Lite is an upper schema for description of resource goal and behaviour. It is based on the CDF schema and extends it as well as the Resource State and Condition Description Framework Schema (RS/CDFS) does [21].

rgbdfs:Goal_Statement is a class of the goal instances. This class is similar to rscdfs:SR_Statement and is its subclass. Triple <SSS-PPP-OOO> describes a statement about a fact, which is currently absent or has "false"

[17] http://www.cs.jyu.fi/ai/papers/JBCS-2005.pdf
[18] http://www.cs.jyu.fi/ai/OntoGroup/

value in the resource history and which a resource aims to have (i. e. an agent must achieve this goal). Each goal is dynamic and can belong to a resource only in certain context (see Figure 2).

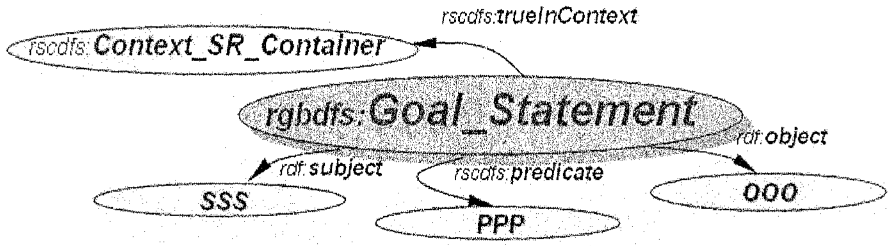


Figure 2. Goal Statement

rgbdfs:Behaviour_Statement - a class of the behavioural instances is represented in Figure 3. This class is a subclass of rscdfs:SR_Statement with extended properties. The rscdfs:ResourceAgent class plays role of a range for the subject (***rgbdfs:subject***). Range of the statement's predicate (***rgbdfs:predicate***) is restricted by the ***rgbdfs:B_Property*** class (subclass of rscdfs:SR_Property).

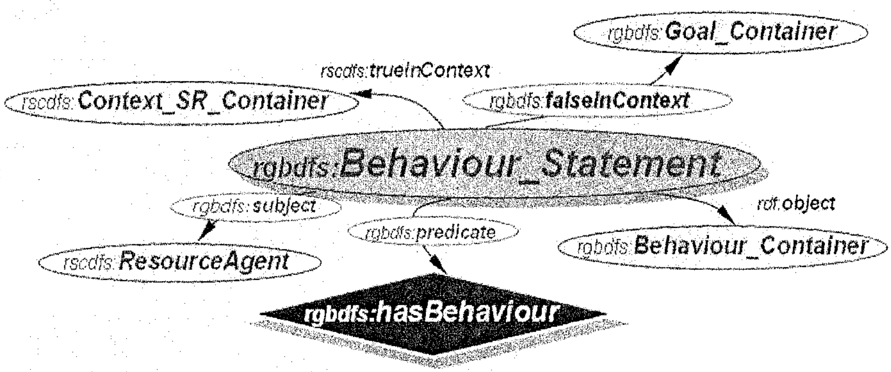


Figure 3. Behavioural Statement

An object of the behavioural statement can be represented by rgbdfs:Behaviour_Container: a container for nested behavioural statements (if a root behaviour is complex) or for atomic execution(s) The ***rgbdfs:falseInContext*** property is a sub property of the rscdfs:falseInContext property. This property has a little bit different meaning than its super property. It plays a role of a trigger, which switches on and switches off the execution of the rule (execution of the behaviour.

which is described via the behavioural container). Behavioural statement will be true, if in the resource history there is no a statement about the fact of the specified goal. The property makes a link to a goal container, which contains goal statement(s) (because it is reasonable to execute behaviour only when a goal has not been achieved). If presence of a goal is a necessary condition for the behaviour, then contextual statements (condition of the environment) are a sufficient condition (which is represented by the contextual container via the rscdfs:trueInContext property).

rgbdfs:Goal_Container is a class of the instances of a goal container. This class is a subclass of rscdfs:SR_Container in general. It represents a container of goal statements, which define the goals. Such container plays a role of a context (using the rscdfs:falseInContext property) for a behavioural statement till the goal is achieved, and that is why it is a direct subclass of rscdfs:Context_SR_Container. The rgbdfs:gMember property is a property redefined from rscdfs:member and which defines instance of the rgbdfs:Goal_Statement class as a member of the container.

As it was mentioned in the previous chapter, goal can be divided into a set of sub goals. Thus a goal container also plays a role of a set of goals, members of which are sub goals of a complex goal. With the purpose of defining a set of sub goals for a complex goal, the property ***rgbdfs:subGoal*** was defined in RG/BDFS-lite (see Figure 4). The domain and range for this property are rgbdfs:Goal_Statement and rgbdfs:Goal_Container classes correspondingly.

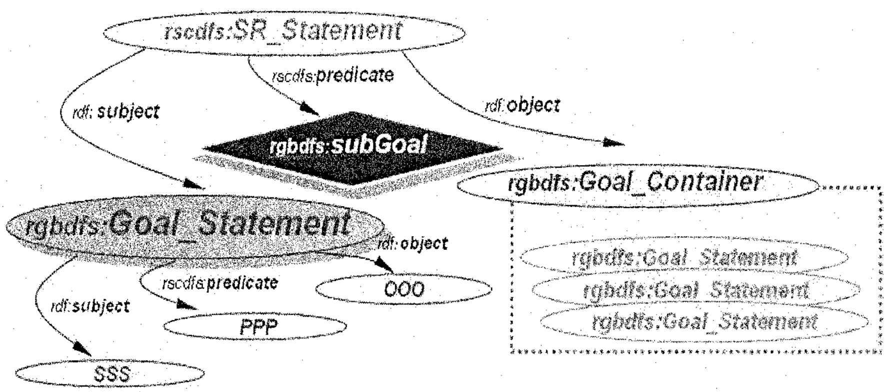


Figure 4. RGBDF Goal

rgbdfs:Behaviour_Container is a class of the instances of a behavioural container. As a subclass of rscdfs:SR_Statement, it has a redefined rgbdfs:bMember property. The main role of the behavioural container is collecting nested behaviours for a complex behaviour (represented by a behavioural statement) (see Figure 5).

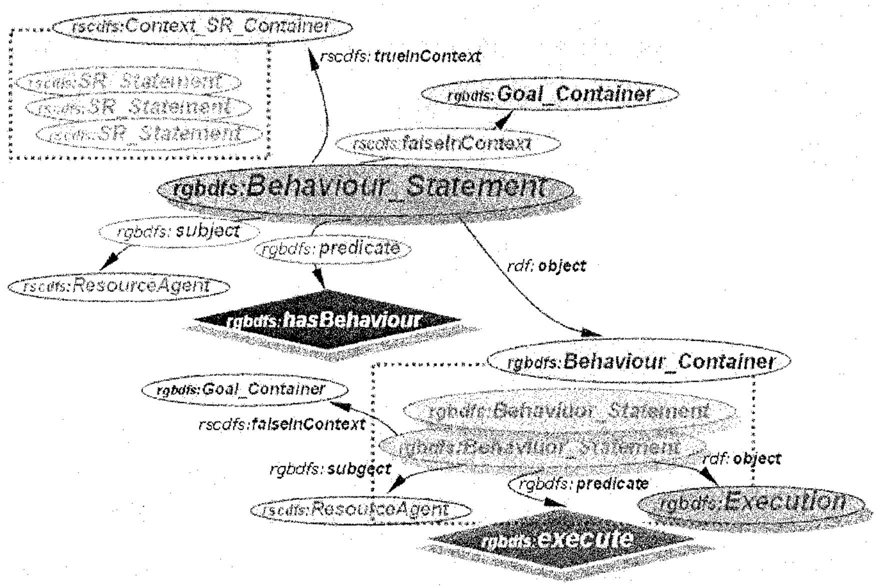


Figure 5. RGBDF Behaviour

Simple behaviour, that assumes execution of a certain action (invocation of certain method, code…), can be described via the ***rgbdfs:execute*** property (instance of the rgbdfs:B_Property class), which defines instance of ***rgbdfs:Execution*** class for a resource agent. This instance describes the exact method (code, service, etc.), inputs, outputs and other features of an executive entity. With a purpose of defining a complex behaviour (which assumes execution of a set of nested behaviours) for an agent, RG/BDFS-lite has the ***rgbdfs:hasBehaviour*** property (instance of the rgbdfs:B_Property class). This property defines a set of behavioural statements for a resource agent via a behavioural container (see Figure 5).

Another important part of behavioural structuring is an agent role (see Figure 6). The ***rgbdfs:hasRole*** property defines a role (***rgbdfs:Role***) for a resource agent in certain context. ***rgbdfs:goals*** is another property related to an agent role and which defines a goal or a set of the goals that correspond to the subject role via a goal container. As it was mentioned previously, resource agent can have different roles, while a set of the goals can be different even for the same role depending on the context (environmental condition). Thus, with a purpose of having a possibility to define a context for them, these two properties are instances of the rscdfs:SR_Property class.

The presented Resource Goal Behaviour Description Framework is fully compliant with the BDI (Belief-Desire-Intention) model well-known in the scientific world of Multi-Agent Systems [16]. By now, a quite big amount of research results is available in the domain of agents based on BDI model.

Recent results report significant development of modelling frameworks for BDI agents [17] and different logic programming languages for it, such as AgentSpeak [18]. BDI model has been actively developed towards cooperative behaviour of agents [19], particularly with a purpose of exchanging executive plans [20]. Figure 7 explains a parallel between BDI and RGBDF models.

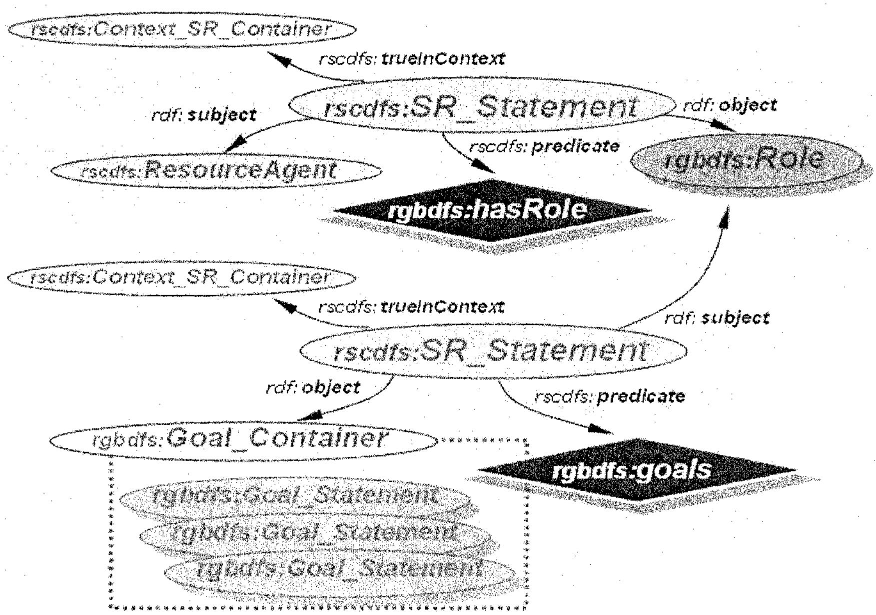


Figure 6. RGBDF Role

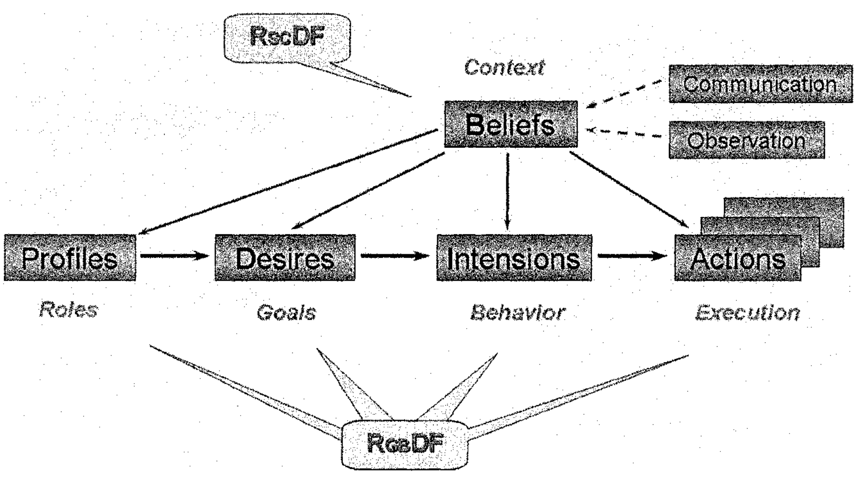


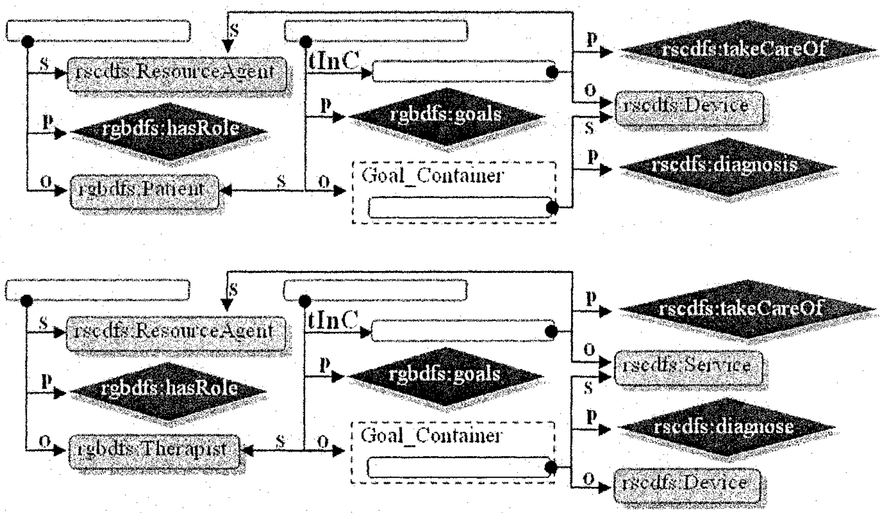
Figure 7. BDI: Underlying Model for RGBDF

4. AGENT BEHAVIOUR CASE

Let us consider a case of agent behaviour. It will be a simple case of a device diagnostics performed by a web service. Actually we have two smart resources: conventional resources (field device and web service) supplied with the agents that maintain them.

Agent, which represents a field device, plays a role of a patient that wants to take self-care (to know its own condition/diagnosis) of itself in case if certain alarm happens. Thus, the goal of this agent is to get a statement about a diagnosis from a diagnostic unit (in our case diagnostic web service) based on sub-history of device states, if an emergency statement appears. It is a complex goal and contains nested sub goals. Agent has to send a diagnostic request to a web service that requires initial collecting of the set of device states and searching appropriate web service. After the request has been sent, the agent must get corresponding response with a statement about diagnosis from the web service. On the other hand we have a web service agent, which plays a role of a therapist (diagnostic unit). The goal of this agent is to diagnose based on sub histories of device states. It is also a complex goal, which assumes receiving a diagnostic request, diagnosing and sending a response back to the field device agent.

As it was mentioned before, ontology contains templates of roles, goals and behaviours. Figure 8 represents two templates of roles and templates of goals that correspond to them.

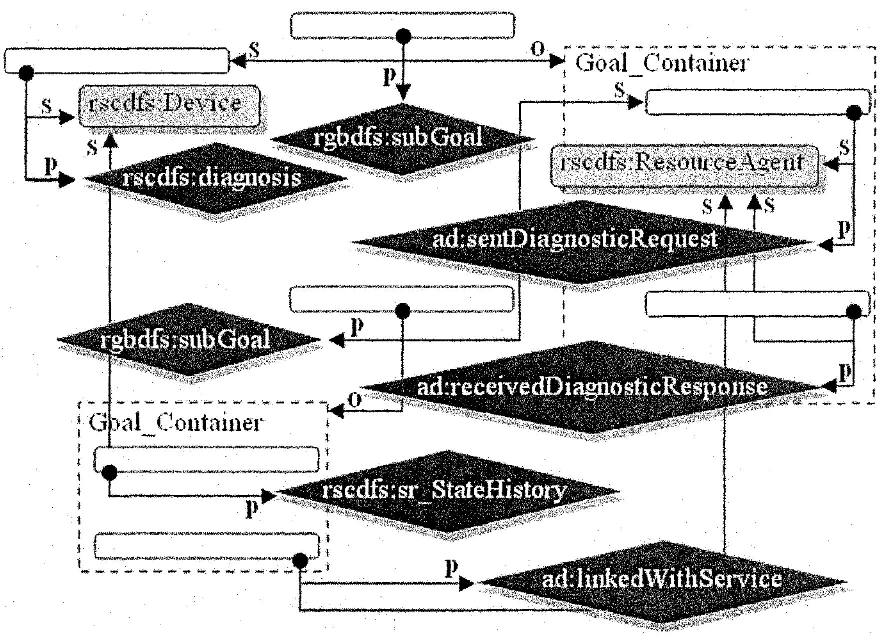


s, **p**, **o**, **tInC** – are *rdf:subject, rscdfs:predicate, rdf:object* and *rscdfs:trueInContext* properties.

Figure 8. Role and corresponding Goal templates

Further, we will concentrate on an example of the first mentioned agent, which maintains a field device. As you can see, the agent, which has a role of a patient, has a goal to get a diagnostic statement about certain device in a context that the agent takes care of this device. However, it is not an atomic goal, and therefore it has a set of nested sub-goals. Next figure demonstrates, how nested sub goals can be described in ontology (Figure 9.).

In a similar way template of agent behaviour can be described via rgbdfs:Behaviour_Statement. Such statement links certain statement about execution (which defines executive module for a certain action through rdf:object property using rgbdfs:Behaviour_Container) with a goal (described through the rscdfs:falseInContext property) and with a context (which specifies a condition, when the action has to be performed through the rscdfs:trueInContext property).



s, p, o – are *rdf:subject, rscdfs:predicate* and *rdf:object* properties.

Figure 9. Representation of Nested Goal

Thus, let us consider a process of goal and behaviour specification for an agent. At the initial stage an expert, which performs linking of the agent to a resource (field device), has to specify from ontology certain role or a goal or even a set of them for current agent. If certain role was specified, then a set of corresponding goals or one goal can be retrieved automatically from the ontolopy. Then, based on goals corresponding to the agent, appropriate behavioural templates also can be retrieved from the ontology. After the all necessary templates are collected, their corresponding instances (with links to concrete instances of the resource agent, resources, etc.) have to be put to the agent storage on the resource platform. Depending on a complexity of the goals, nested hierarchy of agent behavioural rules will be composed automatically by an engine of an agent shell (see the example in Figure 10 and Figure 11).

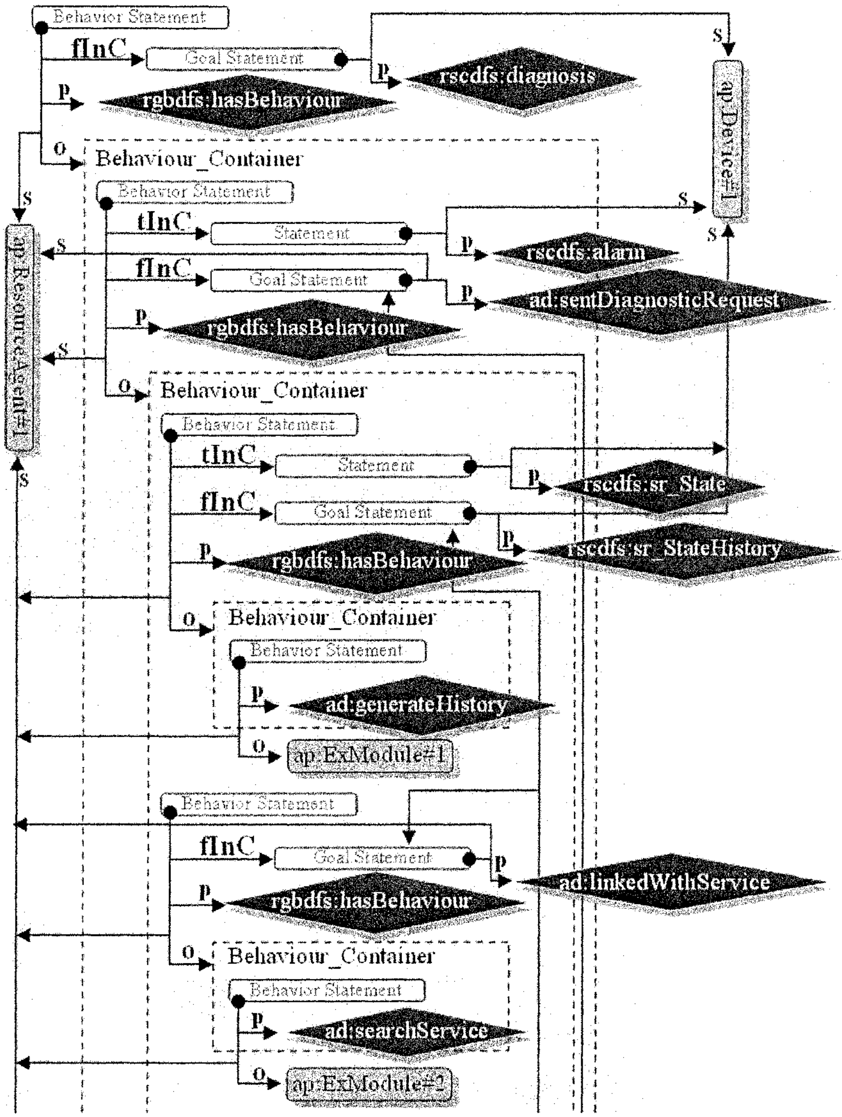
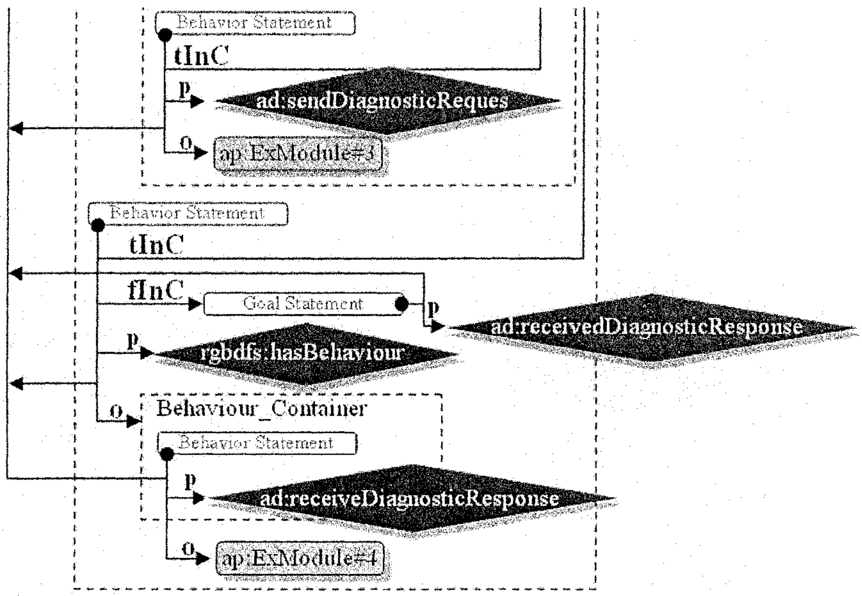


Figure 10. Nested hierarchy of agent behavioural rules



s, **p**, **o**, **tInC**, **fInC** – are *rdf:subject, rscdfs:predicate, rdf:object, rscdfs:trueInContext* and *rscdfs:falseInContext* properties.

Figure 11. Nested hierarchy of agent behavioural rules (continuation)

Note, that actions and behavioural primitives that are stored in a behavioural container, are not assumed to be ordered somehow. The order of their execution is fully defined by context instances, according to which the behavioural primitives are started. This approach provides flexibility in the RGBDF-based modelling.

Now, when the rules of agent behaviour have been specified, it is a time to run agent engine for the behaviour. Working space (storage) of SmartResource Platform (combination of a device and an agent, which maintains it) should be divided into two parts: a temporal storage and a long term one. Initially all information (all statements that concern the states and conditions of resources) are saved to the temporal storage and play a role of a behavioural context and input data for executive modules. As it was mentioned before, the goal statements (linked via the rscdfs:falseInContext property) play a role of a trigger to run certain behavioural rule. If there is no a statement in the temporal storage similar to the goal statement, then the agent engine executes a rule. Let us consider our example. Agent starts to perform root behaviour all the time, when a statement about device diagnosis does not exist in the temporal storage. Otherwise engine skips behavioural rule and passes to the next one on the same level of nesting. Each level can contain both types of behavioural statements: complex behavioural statements and atomic execution statements (which specify executive module via an instance of the rgbdfs:Execution class). In fact, these executive modules generate (add to the temporal storage a statement, which is required by the goal of the behavioural rule). For example, executive module, which is described by the instance *ap:ExModule#3,* generates a statement which asserts that the agent (ap:ResourceAgent#1) has sent a diagnostic request to certain diagnostic service. Thus, agent has achieved one of the sub goals. However, the two sub goals (collecting of a history of the devise states and retrieval of suitable service for the diagnostics) have been achieved before, because they were needed for performing the executive module #3.When the agent has achieved the upper goal, the statement of the achieved goal is moved to the long term storage to be kept in the history of the resource. At the same time all statements of the achieved nested goals have to be removed, too. Some of the contextual statements, which played a role of input data also must be removed (for example, the statement about an alarm, which plays a role of a context for a process of sending a diagnostic request; and statements about the states of the device, which were used for the diagnostics).

5. CONCLUSIONS

As it has been presented, ontology-driven approach in modelling agent behaviour as context-sensitive dynamic change of standardised and reusable roles, goals and actions, anticipated to become a powerful solution providing some benefits comparably to conventional model-driven approaches. RGBDF that was described in this document has been designed within the second stage of the SmartResource project (Proactivity Stage). Resource Goal/Behaviour Description Framework continues development of a modelling basis for the overall SmartResource platform. Further tools and use cases that should be developed within the Proactivity Stage based on RGBDF, will form a ground in favour of ontology-driven approach to modelling proactive resources behaviour.

6. ACKNOWLEDGEMENT

This research has been performed as part of SmartResource project funded by TEKES and industrial consortium of Metso Automation, TeliaSonera, TietoEnator and Science Park.

7. REFERENCES

[1] G. Laleci, Y. Kabak, A. Dogac, I. Cingil, S. Kirbas, A. Yildiz, S. Sinir, O. Ozdikis, O. Ozturk: A Platform for Agent Behaviour Design and Multi Agent Orchestration. AOSE 2004: pp. 205-220.

[2] T. Pirttioja, I. Seilonen, P. Appelqvist, A. Halme, and K. Koskinen, Agent-based architecture for information handling in automation systems, 6th IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services (BASYS 2004), Vienna, Austria, September, 27-29, 2004.

[3] Stollberg, M.; Keller, U.; Zugmann, P.; Herzog, R., Semantic Web Fred - Agent Cooperation on the Semantic Web, 3rd International Semantic Web Conference, Hiroshima, Japan, 7 - 11 November 2004.

[4] Specification of Java library for programming agent behaviours, http://jade.tilab.com/doc/api/jade/core/behaviours/Behaviour.html.

[5] L. Braubach, A. Pokahr, W. Lamersdorf, Jadex: A Short Overview, in: Main Conference Net.ObjectDays 2004, AgentExpo.

[6] M. Griss, S. Fonseca, D. Cowan, R. Kessler, SmartAgent: Extending the JADE agent behaviour model, HPL 2002-18, January 2002.

[7] I. Seilonen, K. Koskinen, T. Pirttioja, P. Appelqvist, and A. Halme, Agent-based approach to enhanced flexibility in process automation systems, 3rd International Symposium on Open Control Systems, Helsinki, Finland, September 9-10, 2003.

[8] Zhao, L. and Mehandjiev, N. and Macaulay, L. (2004) *Agent Roles and patterns for supporting Dynamic behaviour of Web Service Applications.* In: Proceedings of AAMAS04 Workshop on Web Services and Agent-Based Engineering (WSABE).

[9] A. Chella, M. Cossentino, and L. Sabatucci. Tools and patterns in designing multi-agent systems with passi. In *6th WSEAS Int.Conf. on Telecommunications and Informatics (TELE-INFO 04)*, Cancun, Mexico, May 2004.

[10] Djuric, D., Gaševic, D., Damjanovic, V., "AIR A Platform for Intelligent Systems," In Proceedings of the 1st IFIP Conf. on AI Applications and Innovations, Toulouse, France, August 2004.

[11] Viviane Torres da Silva, Ricardo Choren, Carlos José Pereira de Lucena: A UML Based Approach for Modelling and Implementing Multi-Agent Systems. AAMAS 2004: 914-921.

[12] R. Cervenka, I. Trencansky, M. Calisti, D. Greenwood. AML: Agent Modelling Language. Toward Industry-Grade Agent-Based Modelling. Springer LNCS, Vol. 3382/2005: Agent-Oriented Software Engineering V: 5th International Workshop, NY, USA, July 19, 2004.

[13] Dastani, M., Jonker, C.M., and Treur, J., A Requirement Specification Language for Configuration Dynamics of Multi-Agent Systems. *Intern.l Journal of Intelligent Systems*, vol. 19, 2004, pp. 277-300.

[14] Naumenko A., Nikitin S., Terziyan V., Zharko A., Strategic Industrial Alliances in Paper Industry: XML- vs. Ontology-Based Integration Platforms, In: *The Learning Organization*, An International Journal, Emerald Publishers, Vol. 12., No. 5, 2005 (to appear).

[15] Mazzocchi S., It's All about Graphs, White Paper, February 11, 2004, available at: http://www.betaversion.org/~stefano/linotype/news/46/, last accessed 17th April 2005.

[16] Rao, Anand S. and Georgeff, Michael P., BDI Agents: From Theory to Practice, in Proc. of the 1st International Conference on Multi-Agent Systems (ICMAS-95), San Francisco, USA, June, 1995.

[17] Mascardi, V., M. Martelli, F. Zini, and R. Degl'Innocenti: 'CaseLP: a Prototyping Environment for Heterogeneous Multi-Agent Systems'. Journal of Autonomous Agents and Multi-Agent Systems, 2004.

[18] Rao, A. S. 1996. AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language. In Proc. MAAMAW-96. Springer Verlag, LNAI 1038, pp. 42–55.

[19] Ancona D. and Mascardi V., Coo-BDI: Extending the BDI Model with Cooperativity, In Proc. of the First Declarative Agent Languages and Technologies Workshop, J. A. Leite, A. Omicini, L. Sterling and P. Torroni (Eds.), Springer LNAI 2990, 2004, pp. 109-134.

[20] Ancona D., Mascardi V., Hübner J. F., Bordini R. H., Coo-AgentSpeak: Cooperation in AgentSpeak through Plan Exchange, In Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2004), N. R. Jennings and C. Sierra and L. Sonenberg and M. Tambe (Eds.), pp. 698-705, ACM press, 2004.

[21] Kaykova O., Khriyenko O., Naumenko A., Terziyan V., Zharko A., RSCDF: A Dynamic and Context-Sensitive Metadata Description Framework for Industrial Resources, In: *Eastern-European Journal of Enterprise Technologies,* Vol. 3, No. 3, June 2005, ISSN: 1729-3774, available in: http://www.cs.jyu.fi/ai/RSCDF-2005.doc

[22] Terziyan V., Semantic Web Services for Smart Devices Based on Mobile Agents, In: *International Journal of Intelligent Information Technologies*, Vol. 1, No. 2, April-June 2005, Idea Group, pp. 43-55.

[23] Kaykova O., Khriyenko O., Kovtun D., Naumenko A., Terziyan V., Zharko A., General Adaptation Framework: Enabling Interoperability for Industrial Web Resources, In: *International Journal on Semantic Web and Information Systems*, Vol. 1, No. 3, 2005, pp. 30-62.

THE STUDY ON THE SEMANTIC IMAGE RETRIEVAL USING THE COGNITIVE SPATIAL RELATIONSHIPS IN THE SEMANTIC WEB

Hyunjang Kong[1], Myunggwun Hwang[1], Kwansang Na[3] and Pankoo Kim[2]
[1]Dept. of Computer Science, Chosun University, Gwangju 501-759, Korea; [2]Corresponding Author, Dept. of Computer Engineering, Chosun University, Gwangju 501-759, Korea; [3]Dept. of Computer Science and Engineering, Korea University, Seoul 136-701, Korea

Abstract: In present day, there are a number of image data in the web because of the development of the image acquisition devices. So, many researchers have been studying about the image retrieval and management. Keyword matching, contents-based and concept-based methods are the basic studies for the image retrieval. In this paper, we suggest the new image retrieval methodology using the cognitive spatial relationships between the objects in the image. There were the similar studies already using the spatial relationships. However, the studies have the limitations and don't give the good search results. We think to need the new methodology for representing the spatial relationships. It is the cognitive spatial relationships. In our study, we newly define the cognitive spatial relationships and apply it to the image retrieval system. At the result, we realized that our methodology makes possible the semantic image retrieval.

Key words: Cognitive Spatial Relationships, Ontology, Semantic Web, Image Retrieval

1. INTRODUCTION

There are a huge number of data in the web. Nowadays, users want to search the information more rapidly and correctly. Until now, most information is the document types so the methodology for the information retrieval is based on the text matching.[2][3][4] The methodology provides the sufficiently correct information to users from the web. However, as the image acquisition technologies such as digital camera, a scanner, cellular phone camera, etc, have improved many images have being existed in the web environment. And the existing retrieval system just using the keyword matching about the image has the limitation. Because most images have the contents, they are generally stored with simple annotations. Thus, many researchers have been studying the methodologies for image retrieval. The basic methodologies studied until now are the contents-based, concepts-based and ontology-based methodologies.[5][6][7] However, the methodologies still don't give the good results to the users. So, we suggest the new methodology for the semantic image retrieval in this paper. Our idea is the cognitive spatial relationships. We focus on the spatial relationships and we define the cognitive spatial relationships. And then we build the spatial ontology based on the cognitive spatial relationships, user's research, WordNet and OXFORD dictionary. Finally we designed the new image retrieval system using the spatial ontology.

In the 2nd chapter, we introduce the related works – information retrieval system, semantic ontology-based image retrieval system and spatial description logics. Then in chapter 3, we explain the background knowledge of our study and the cognitive spatial relationship, and how to build the spatial ontology. In chapter 4 and 5, we describe our system based on the cognitive spatial relationship and experimental results and evaluation of our system. In the end of this paper, we conclude our study and suggest the future works.

2. RELATED WORKS

2.1 Information Retrieval System

Information Retrieval systems are used to store, maintain, search, and retrieve the information items. The information items could be text documents, images, sounds or videos. In the information retrieval system, it is very important to have efficient data structures, fast search tools, and effective information retrieval methods, especially if the amount of data is large. Generally, most information retrieval systems utilize indexing methods to improve the search efficiency. Indexing is the process of assigning descriptive terms to information items for retrieval purposes. Indexing is a very important and difficult task and every information item is stored in a traditional information retrieval system with the index. However, documents often lose their semantics when represented by just simple index terms. Therefore, normally search of the documents using simple keywords results in retrieving irrelevant documents, which is the case with most web search engines.[8]

2.2 Ontology-based Image Retrieval

As mentioned in above section, the traditional information retrieval systems have the mismatch problem between the terminologies. For solving the problem, many researchers have studied to apply the ontology theory. A great many works show that ontologies could be used not only for annotation and precise information retrieval [9], but also for helping the user in formulating the information need and the corresponding query. This is important especially in applications where the domain semantics are complicated and not necessarily known to the user. Furthermore, the ontology-enriched knowledge base of image metadata can be applied to constructing more meaningful answers to queries than just hit-lists.

The major difficulty in the ontology-based approach is the extra work needed in creating the ontology and the detailed annotations. We believe, however, that in many applications this price is justified due to the better accuracy obtained in information retrieval and to the new semantic browsing facilities offered to the end-user. We are trying to implement semantic techniques to avoid so much hard work with the ontology building--the trade--off between annotation work and quality of information retrieval can be balanced by using these less detailed ontologies and annotations. Although this approach could address the mismatch problem between the terms, it is still not suitable for image retrieval system because they did not consider the features of the image data. Therefore, we will not get good results in the ontology-based image retrieval system.

2.3 The Description Logic $\mathcal{ALC}$(Drcc8)

The *Region Connection calculus* *RCC*-8[1] is a language for qualitative spatial representation and reasoning where the spatial regions are regular subsets of a topological space. The regions themselves do not need to be internally connected i.e. a region may consist of different disconnected pieces.

As the concrete domain in $ALC(\mathrm{D}_{recc8})$, $\Delta^{D}{}_{rcc8}$ is the set of all the non-empty regular closed subsets of the topological space R^2. $\Phi^{D}{}_{rcc8}$ is obtained by imposing a union, intersection, composition and converse operations over the set of the elementary binary relationships between the regions i.e.(PO, NTPP, TPP, EQ, TPP^{-1}, NTPP^{-1}, EC, DC) where the intended meaning of the elements are respectively Proper Overlap, Non Tangential Proper Part, Tangential Proper Part, External Connection, and DisConnected.

3. OUR APPROACH

3.1 Background Knowledge of the Cognitive Spatial Relationships

In the existing image retrieval system, if the system uses the spatial relationships between object, the system firstly extracts the edge of the objects. Secondly, the system use the spatial relationships based on the regions of the objects. In such a case, the biggest problem is either the spatial relationships do not have the semantic meaning or system defines the spatial relationships incorrectly. Figure 1 explains the general spatial relationships using in the existing image retrieval system.

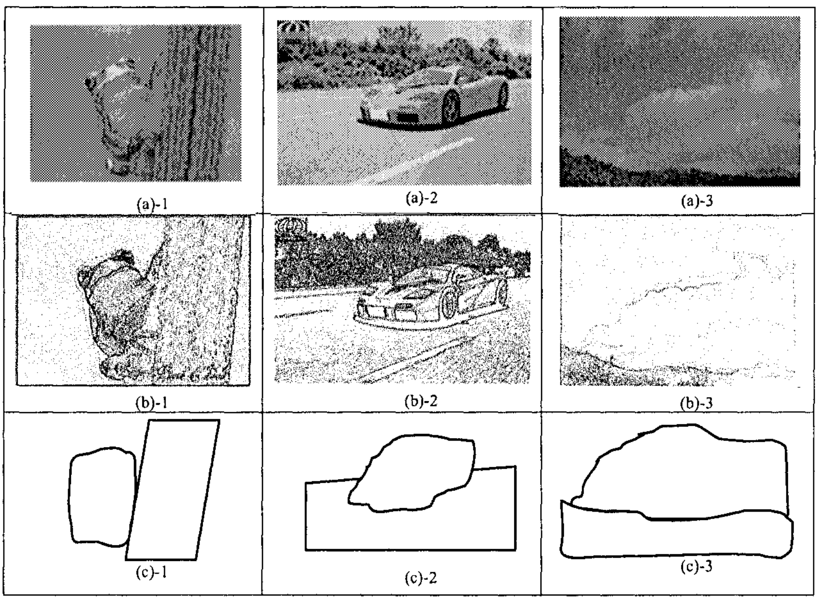


Figure 1. (a) images are the original images, (b) images are the edges of the objects, (c) images are the regions of the objects

In above three images, if the system represents the spatial relationships, it is like as:

- *First images((a)-1,(b)-1,(c)-1) are represented the 'flog-connected-tree'.*
- *Second images((a)-2,(b)-2,(c)-2) are represented the 'car-part of-road'.*
- *Third images((a)-3,(b)-3,(c)-3) are represented the 'mountain-connected-cloud'.*

However, the users recognize different with the system. The cognitive spatial relationships recognized by users are like as:

- *First image((a)-1) means the 'flog-connected-tree'*
- *Second image((a)-2) means the 'car-connected-road'*
- *Third image((a)-3) means the 'cloud-disconnected-mountain'*

Nowadays, the users want that the machine also thinks and processes like the human. It is the basic idea of the semantic web. In the semantic web environment, users make search requests for images based on their visual impressions. If the system stores the image metadata using the region-based spatial relationships, the system will provide the wrong and senseless results to users. Therefore, we try to define the cognitive spatial relationships newly and design the image retrieval system based on our study.

3.2 Definition of the Cognitive Spatial Relationships and Construction of the Spatial Ontology

In our study, we used the research for defining the cognitive spatial relationships. We prepare the 200 images containing the objects and spatial relationships between them. And then, we examine the spatial relationships recognized by users when users look at the images. At the result of the research, the cognitive spatial relationships represent the basic three kinds of relationships. Throughout the research, we realized that most images are represented by the 'connect', 'disconnect' and 'partof' relationships. Figure 2 illustrates the model of the cognitive spatial relationships comparing with the *RCC-8*.

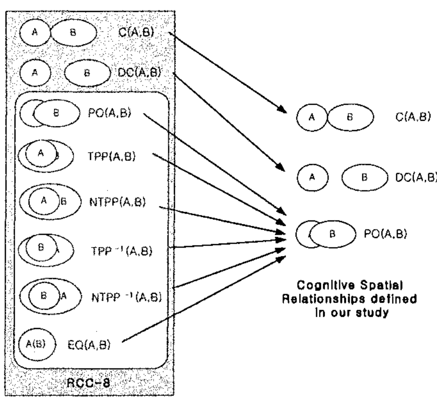


Figure 2. The model of the cognitive spatial relationships

In our study, another significant feature is to build the spatial ontology based on the cognitive spatial relationships and user's research. For grasping the spatial verbs, we examine them using the experimental document containing the sample images. The contents and results of the research are like as:

image #3			image #33	image #40		image #55	image #59		image #141
Research results about the images(spatial relationships 1)connect 2)partof 3)disconnect, C_SR:cognitive spatial relationships, sp:spatial relationships)									
number	C_SR	sp	Answers of the Researchers						
3	1	2	lying	lying	lie	lying	lying	lie	sit
33	2	2	swim	search	swim	swim	cross	look pretty	swim
40	1	1	kiss	kiss	kiss	kiss	kiss	love	play
55	3	3	on the left	bigger	beside	behind	left of	beside	beside
59	3	2	wait	in front	waiting	stand	in front	look	stand
141	2	2	fly	soars	fly	fly	fly	fly	fly

Figure 3. The parts of the Experimental Document

In the results of the research, we realized that most users have the similar feeling and use the similar spatial verbs to represent the images.

Thus, we build the spatial ontology based on the cognitive spatial relationships and spatial verbs. The most important parts in the spatial ontology are the cognitive spatial relationships defined in section 3.1 and the spatial verbs. And we adopt the WordNet and OXFORD Dictionary to make more semantic ontology. Figure 4 shows the architecture of the spatial ontology based on the cognitive spatial relationships.

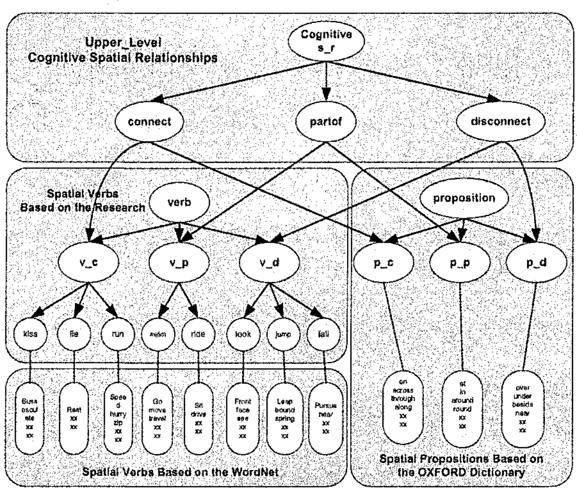


Figure 4. The architecture of the spatial ontology

In figure 4, the cognitive spatial relationships are situated in the top level and the second level consists of the two parts – spatial verbs and spatial propositions. The bottom level is added by the instances based on the WordNet and OXFORD Dictionary. The significant fact newly known throughout the research is that not only the verbs but also the propositions are very important to present the cognitive spatial relationships. Therefore, we made the spatial verbs part based on the WordNet and the spatial propositions part using the OXFORD Dictionary. The table 1 and 2 show the instances used in spatial ontology based on the WordNet and OXFORD Dictionary.

Table 1. Research words match with the WordNet words

Cognitive spatial relationships	**Research words**	**WordNet matching words**
connect	Attach	Connect, link, tie, link up, fasten, touch, adjoin, meet, contact
connect	Kiss	Buss, osculate
disconnect	Chase	Chase after, trail, tail, tag, give chase, god, go after, pursue, follow
disconnect	Jump	Leap, bound, spring
partof	Float	Drift, be adrift, blow, swim, transport
partof	Hide	Conceal, shroud, enshroud, cover, obscure, blot out, obliterate, veil

Table 2. The spatial propositions

The spatial propositions based on OXFORD Dictionary	
connect	On, along, across, through
disconnect	Over, under, above, below, by, beside, near, before, behind
partof	At, in, around, round

In our study, firstly we defined the cognitive spatial relationships, and secondly built the spatial ontology based on the cognitive spatial relationships. The cognitive spatial relationships are written by the ontology language for applying to the image retrieval system. In the image retrieval system, the super user can store the image data using three cognitive spatial relationships and the end user can search the image accessing the spatial ontology. We designed the system to be able to query using the natural language. We serialize the spatial ontology using OWL and table 3 shows the OWL syntax about the spatial ontology.

Table 3. The part of the spatial_ontology.owl

```
.........
<owl:Class rdf:ID="CognitiveSpatialRelationship">
  <rdfs:subClassOf rdf:resource="#SpatialRelationship" />
</owl:Class>
<owl:Class rdf:ID="connect">
  <rdfs:subClassOf rdf:resource="#CognitiveSpatialRelationship" />
</owl:Class>
 .........
<owl:Class rdf:ID="verb">
  <rdfs:subClassOf rdf:resource="#word" />
</owl:Class>
<owl:Class rdf:ID="verbConnect">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#verb" />
    <owl:Class rdf:about="#connect" />
  </owl:intersectioinOf>
</owl:Class>
<owl:Class rdf:ID="Kiss">
  <rdfs:subClassOf rdf:resource="#verbConnect" />
</owl:Class>
<Kiss rdf:ID="buss" />
<Kiss rdf:ID="osculate" />
 .........
<owl:Class rdf:ID="proposition">
  <rdfs:subClassOf rdf:resource="#word" />
</owl:Class>
<owl:Class rdf:ID="propositionConnect">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#proposition" />
    <owl:Class rdf:about="#connect" />
  </owl:intersectioinOf>
</owl:Class>
<propositionConnect rdf:ID="on" />
<propositionConnect rdf:ID="across" />

.........
```

4. THE IMAGE RETRIEVAL SYSTEM APPLYING THE COGNITIVE SPATIAL RELATIONSHIPS

Our system consists of three parts.

- Content provider interface part – Content provider stores and manages the images
- End user interface part – End user retrieval the images
- Ontology part – Domain and spatial ontologies are in this part.

Figure 5 illustrates the architecture of our system.

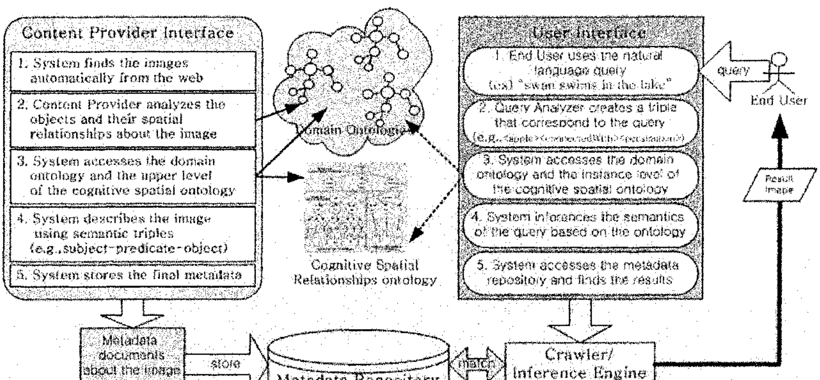


Figure 5. The architecture of the experimental image retrieval system

Our system has two significant features. One is the application of the spatial ontology constructed based on the cognitive spatial relationships. Another is the capability to process the user query using the natural language. We expect improvement of semantic image retrieval throughout our study.

5. EXPERIMENTAL RESULTS AND EVALUATIONS

For evaluating our system, we test our system using several test beds. The test systems are Google, Yahoo and our system. Google and Yahoo use the big category that is the kind of the ontology. The sample queries for testing are like as:

1. *Only one word query – e.g. swan*
2. *Two words query – e.g. swan and lake*
3. *Query containing the spatial relationships – e.g. swan in the lake*
4. *Natural Language query containing the spatial verbs – e.g. swimming swan*
5. *Natural Languages query containing the spatial verbs and proposition – e.g. swan swims in the lake*

For testing, we prepared the related sample images. We measure the precision of the search results in three test systems. Because three systems have the different image resources, we measure the accuracy of each system. For measuring the accuracy, we use the simple formula showing as below:

$$\text{Accuracy} = \frac{\text{Correct images matched with the query}}{\text{All images searched throughout the system}}$$

Table 4 shows the result about the test.

Table 4. Expremental results

	1	2	3	4	5
Google	27/200 : 0.135	20/200 : 0.100	67/200 : 0.335	10/30 : 0.333	25/52 : 0.481
Yahoo	72/200 : 0.360	17/48 : 0.354	49/200 : 0.245	7/25 : 0.28	12/26 : 0.462
Our System	42/109 : 0.385	37/60 : 0.617	37/39 : 0.949	35/37 : 0.946	35/35 : 1.000

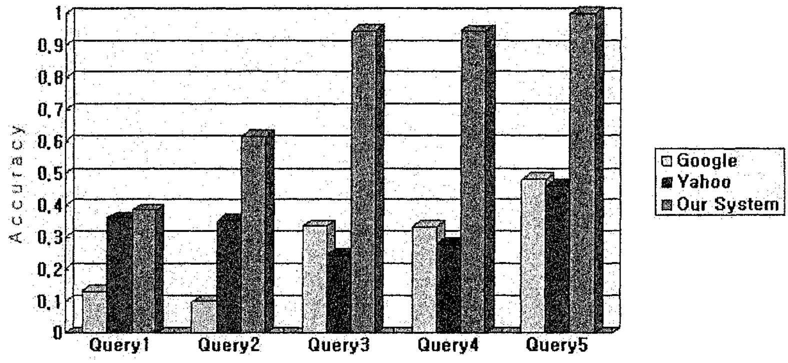


Figure 6. The graph representation of the experimental results

In table 4, we realized that there is no big difference between our system and other systems about the simple text query – query 1 and query 2. However, we could know that our system give the excellent result about the complex queries – query 3, 4 and 5 comparing with other systems. At the result, we can approach more semantic image retrieval and the natural language query processing. Figure 7 and Figure 8 show the results of our system about the test query 4 and 5.

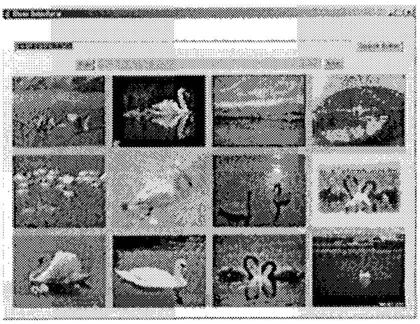


Figure 7. The result about the query 4

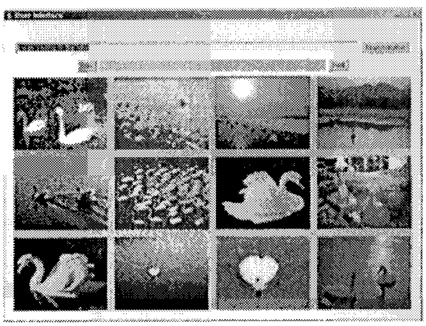


Figure 8. The result about the query 5

6. CONCLUSION

The main features of our study are the definition of the cognitive spatial relationships and construction of the spatial ontology using the spatial verbs and propositions. Thus, we realized that the natural language query processing and the semantic image retrieval are possible based on our idea.

However, we have also the limitation that the content provider needs much time to annotate the image semantically throughout our system. It remains our future study. In conclusion, our study presents the vision of the semantic image retrieval and the natural language query processing.

ACKNOWLEDGEMENT

This study was supported(in part) by research funds from Chosun Unversity, 2003.

REFERENCES

1. Deborah L. McGuinness, Daniele Nardi, Peter F.Patel-Schneider, "The Description Logic Handbook: Theory, implementation, and applications", Cambridge University Press, January 2003.
2. Arnold W.M. Smeulders, "Content-Base Image Retrieval at the End of the Early Years", IEEE Transactions on pattern analysis and machine intelligence, Vol. 22, No. 12, December, 2000.
3. Ching-chih Chen, and James Z. Wang, "Large-scale Emperor Digital Library and Semantics-sensitive Region-based Retrieval", Proceeding of the International Conference on Digital Library -- IT Opportunities and Challenges in the New Millennium, National Library of China, Beijing, China, pp. 454-462, July 9-11, 2002.
4. Huamin Feng, Rui Shi, Tat-Seng Chua, "A bootstrapping framework for annotating and retrieving WWW images", Proceedings of the 12th annual ACM international conference on Multimedia, October 2004.
5. Von-Wun Soo, Chen-Yu Lee, Chung-Cheng Li, Shu Lei Chen, Ching-chih Chen, "Automatic metadata creation: Automated semantic annotation and retrieval based on sharable ontology and case-based learning techniques", Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries, May 2003.
6. HuaMin Feng, Tat-Seng Chua, "Image retrieval: A bootstrapping approach to annotating large image collection", Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval, November 2003.
7. Meng Yang, Barbara M. Wildemuth, Gary Marchionini, "Technical poster session 1: multimedia analysis, processing, and retrieval: The relative effectiveness of concept-based versus content-based video retrieval", Proceedings of the 12th annual ACM international conference on Multimedia, October 2004.
8. Carmine Cesarano, Antonio d'Acierno, Antonio Picariello, "Intelligent web information access: An intelligent search agent system for semantic information retrieval on the internet", Proceedings of the 5th ACM international workshop on Web information and data management, November 2003.
9. G. Schreiber, I. Blok, D. Carlier,W. van Gent, J. Hokstam, and U. Roos. A mini-experimentin semantic annotation. In I. Horrocks and J. Hendler, editors, The Semantic Web – ISWC2002. First international semantic web conference, number LNCS 2342, pages 404–408.Springer–Verlag, Berlin, 2002.

TOWARDS CSPACES: A NEW PERSPECTIVE FOR THE SEMANTIC WEB

Francisco Martín-Recuerda
Digital Enterprise Research Institute (DERI)
Leopold-Franzens Universität Innsbruck, Austria

francisco.martin-recuerda@deri.org

Abstract: Information overload is mainly the result of the combination of four factors: the enormous amount of information available; the heterogeneity of information sources and information channels; the generation of a significant percentage of redundant information; and inefficient mechanisms for filtering, searching and classifying information. Given that the former factor cannot be changed, and current forecast expects that information grows exponentially in the next years, research and industry efforts are focusing to overcome the other three. The association of machine-understandable semantics to formally describe data published on the Web and the development of appropriate tools that can handle this method to describe data are the approaches that the promoters of the Semantic Web have suggested to overcome the problem of information overload in the Web. Although, the Semantic Web promises a new level of service with regard to the current Web, a more drastic approach is required. Conceptual Spaces (CSpaces) envision the future of the Semantic Web as a cooperative environment where communication between humans, machines, and human-and-machines will be reduced to the acts of publishing and reading machine processable semantics in a persistent collection of individual and shared information spaces. Decreasing the amount of syntactic data representation in the Semantic Web, and therefore, make machine processable semantics the prevalent representation formalism will facilitate interoperation between heterogeneous applications, web services, agents, humans and so on. Natural language generation and graphical knowledge visualization techniques will make possible that humans deal with this "purest semantic" Web. In addition, CSpaces will also decrease redundancy of the information stored and will provide a better organization of the data articulated around ontologies.

Key words: Information overload, Semantic Web, metadata, Triple Space Computing, Event-based, Ubiquitous Computing, Peer-to-Peer, Tuple Space, personal and distributed knowledge management.

1. INTRODUCTION

In the past decades the situation of a shortage of accessible information has been gradually changing thanks to new communication means like electronic mail and the World Wide Web. In fact, the exponential growth of the information available (e.g. in 2002 our society produced around 5 exabytes[1] of new information mostly stored in hard disks [1]) distributed in several heterogeneous information channels (i.e. emails, fax, instant messages, web pages, etc) have increased the difficulty to organize, find, integrate and reuse this flow of data. Furthermore, these information channels are not coordinated and generate a significant amount of redundant information. The combination of huge amount of information, diversity of information channels, a significant amount of redundant information, and inefficient mechanisms for filtering, searching and classifying information contribute to the appearance of a new phenomenon difficult to overcome: information overload. One approach to alleviate this situation in the concrete scenario of the Web came from the W3C, and it is called Semantic Web [2]. The Semantic Web stands for the idea of a future Web which aims to increase machine support for the interpretation and integration of information. Annotating the Web with formal semantic descriptions together with domain theories (i.e., ontologies) will enable a Web that provides more effective discovery, automation, integration, and reuse of the information currently stored [3]. Unfortunately, the development of the Semantic Web does not deal with the problems of diversity of information channels and redundancy. Moreover, the Semantic Web is an ongoing research effort where several relevant questions are still open. In the scope of this paper, how to physically and scaleably organize and store metadata in a global scenario, how to restrict the access to this metadata, how to guarantee trustworthiness of the available metadata, and how to facilitate the encoding, access and visualization of semantic data representations for human actors, are my major concerns. Also, the dichotomy of Web data (syntactic-textual representation) and semantic annotations will require permanent significant efforts to maintain coherence on both sides and will not reduce the problem of data-redundancy on the Web side.

Information overload can be achieved using a platform based on semantic web technologies that takes into account the following guidelines:

- Unify information channels and sources in a suitable one that can be shared by humans and machines.
- Increase machine support for information management by making machine processable semantics the prevalent representation formalism in this new infrastructure.
- Reduce heterogeneity by introducing mechanisms that allow humans to find agreements in the representation and definition of common terminology and semantic data specifications.
- Promote the use of ontologies as articulation means for organizing data and identifying redundant data.

Conceptual Spaces (CSpaces) envision the future of the Semantic Web as a cooperative environment where communication between humans, machines, and human-and-machines will be reduced to the acts of publishing and reading machine processable semantics in a persistent collection of individual and shared information spaces. Applications and humans can be owners of several Individual CSpaces and can be members of several shared CSpaces (preferably one for each Individual Cspace). Access rights can be easily associated to CSpaces, and interoperability can be improved through the use of Shared CSpaces. Decreasing the amount of syntactic data representation in the Semantic Web, and therefore, make machine processable semantics the prevalent representation formalism will facilitate interoperation between heterogeneous applications, web services, agents, humans and so on. Natural language generation and graphical knowledge visualization techniques will make possible that humans deal with this "purest semantic" Web.

The evolution of present Semantic Web proposal into CSpaces can increase the range of applications and benefits for industry, research and education areas. In particular, I am investigating the applicability of CSpaces in personal and distributed knowledge management, enterprise application integration, Semantic Web services, software components coordination and ubiquitous computing.

The paper is organized as follows. Section 2 provides a short overview of current state of the art and open issues on related Semantic Web technologies. Section 3 describes some relevant building blocks that will constitute Conceptual Spaces. Section 4 discusses some possible applications of this technology. Finally, conclusions and future work are provided in section 4.

2. STATE OF THE ART IN RELATED SEMANTIC WEB TECHNOLOGIES

CSpaces have been invented to minimize the problem of information overload, and as a response to the questions: how to organize and store metadata, how to restrict the access to this metadata, how to guarantee trustworthiness of the available metadata, and how to facilitate the encoding, access and visualization of semantic data representations for human actors, that Semantic Web does not address yet. In this section, a review of the state of the art of relevant Semantic Web technologies is described and significant gaps are identified. This analysis will drive to the necessity to develop a new proposal for organizing, protecting and sharing machine processable semantics.

2.1 Machine processable semantics organizational model

There is not agreement in the Semantic Web community about how to organize ontologies, instances, rules and mappings (alignments) between them. [4] suggests an hybrid approach called ontology islands, where ontologies are mapped to concrete influential domain ontologies , where within the island there is a form of global integration; one ontology would be the global ontology of the islands and a number of local ontologies are mapped to this global ontology. Unfortunately this approach mainly focuses in the mapping problem, and more general model is required.

HCONE [5]introduces three different spaces in which ontologies can be stored: Personal Space, Shared Space and Agreed Space. Ontologies are created in Personal Spaces and shared in Shared Spaces where users can discuss ontological decisions. When users reach an agreement, ontologies are moved to Agreed Spaces. Although closely related with CSpace approach, HCONE is more oriented for collaboratively building ontologies in a restricted community of users than for a Semantic Web scale organizational model.

Outside of the Semantic Web, [6, 7] proposes a distributed knowledge management infrastructure based on Kpeers. An interesting point of this solution is that knowledge is created and shared using a bottom-up approach. Users create their own knowledge that they share with other users, creating bigger knowledge bases.

Finally, CO4 (Collaborative construction of consensual knowledge bases) [8] is an infrastructure enabling the collaborative construction of a knowledge base through the web. The main contribution of this approach is a proposal for organizing KBs in a tree structure. The leaves are called user KBs, and the intermediate nodes, group KBs. Each group knowledge base represents the knowledge consensual among its sons (called subscriber knowledge bases). This solution is closely related with the idea of ontology islands presented before, and it can be easily adapted as a organizational model of machine processable semantics for the Semantic Web.

2.2 Coordination model for distributed systems

Middleware is the "glue" that facilitates and manages the interaction acroos heterogeneous computing platforms. During the last three decades, several coordination models and infrastructures like RPC [9], Message passing, Message Queues [10], Tuple Space [11] and Publish- Subscribe [12] bring the attention of developers of concurrent and distributed systems. Currently, major efforts are focused on Web Services[2] and adding semantics to Web Services[3]. However, [3] and [13] criticize the use of message passing paradigm for Web Services because does not rely in Web principles (asynchronous publish and read model). Thus, they propose an alternative coordination model, called Triple Space, that enriches Tuple Space model using RDF triples instead of unformatted tuples. Parallel work published in [14] reached the same conclusion.

[12] identifies three desirable orthogonal dimensions for coordination models that are extended to four in [15]:

- Space decoupling: processes involved in the interaction can run in completely different computational environments.
- Reference decoupling: processes involved in the interaction do not need to know each other (anonymous).
- Time decoupling: processes do not need to be up at the same time during the interaction (asynchronous).
- Flow decoupling: main flows of process are not affected for the generation of reception of data (no blocking read (receive) and write (send) operations).

Table 1 (partially adapted from [12]) summarizes decoupling dimensions of five coordination models.

Table 1. Decoupling dimensions of several coordination models

Abstraction	Space	Time	Reference	Flow
Message passing	Yes	No	No	Producer-side
RPC	Yes	No	No	Producer-side[4]
Message queues	Yes	Yes	Yes	Producer-side
Tuple Space	Yes	Yes	Yes	Producer-side

[2] http://www.w3.org/2002/ws/

[3] http://www.daml.org/services/, www.wsmo.org and http://lsdis.cs.uga.edu/projects/meteor-s/

[4] In the case of RPC is not clear which process is producer and which is consumer. [9] suggests that consumer plays the role of invoker and producer lays the role of invokee.

Abstraction	Space	Time	Reference	Flow
Publish - Subscribe	Yes	Yes	Yes	Yes

[15] identifies some drawbacks in Tuple/Triple Space. For instance it does not provide flow decoupling dimension, and the model expects a tacit agreement between producers and consumers regarding the format of the data that it will be published in the space (in other words, there is no way to know before hand which is the format of the data that producers will publish information in the space, and therefore, there is no way to know which data format the consumers expect). [15] proposes a new model that combines main features of Tuple/Triple Space and Publish-Subscription models. Further on, I provide a detailed description of this paradigm and how can become in the coordination model for CSpaces. As a consequence of this decision, the architecture proposal for CSpaces has to take into account the requirements of this new coordination mechanism.

2.3 Semantic interoperability

Data integration approaches aim to provide a unified (or reconciled) view, called global or mediated schema of different heterogeneous data sources (local schemas) [16]. Two basic approaches have been used to specify the mapping between local schemas and global schema: global-as-view (GAV) and local-as-view (LAV). The first approach, defines the global schema in terms of the local schemas. In the second approach, the global schema is defined independently from the local schemas, and those local schemas have associated a description of themselves in terms of the global schema. Examples of the GAV approach are TSIMMIS [17], InterViso [18] and Garlic [19] while examples of the LAV approach are IM [20] and Agora [21].

[22] reports that the major disadvantage of GAV is that it is complex and expensive to support evolution of local schemas. On the other hand, it is also well known that processing queries using LAV approach is a difficult task ([22] and [23]).

[24] and [16] propose alternative approaches to combine the advantages of GAV and LAV. The former, called "both-as-view" (BAV), is built on top of a low level hyper-graph-based data model (HDM) and a set of primitive schema transformations for this model. The second approach proves that its is possible to derive LAV view definitions in to GAV view definitions in very restricted scenarios where views are expressed as conjunctive queries, and the global schema is defined in the relational data model with inclusion dependencies.

The Semantic Web devises a more complex scenario for data integration where LAV, GAV and BAV can also be applied [25] and [26]. Ontologies define a more sophisticated schema specifications base on rich formal representation languages. Given the fact that reasoning tasks over expressive languages are very expensive in terms of computational resources, and scenario of mapped heterogeneous ontologies would drive to unacceptable time response of inference systems. Moreover, there would be cases in which semantic heterogeneity of several mapped ontologies cannot be completely reconciliated, and thus inconsistent problem can arise ([25] and [27]). Deal with inconsistencies would require the use of concrete techniques that can degrade even more performances of inference systems.

[27] discusses an alternative approach to deal with queries for mapped heterogeneous ontologies. The authors propose the generation of tailored reasoning spaces for each of the queries that the system receives. The drawback of this approach is that the time necessary to generate such space can be more expensive that perform the query over the relevant mapped ontologies using query rewriting techniques. However, the idea of improving reasoning performance by creating reasoning spaces is one of the motivations because I propose to organize data semantics in the Semantic Web in individual and shared conceptual spaces.

2.4 Knowledge visualization and natural language generation (NLG)

Knowledge visualization comprises all the techniques and mechanisms that facilitate the exploration and visualization of semantic formal representation of information stored in knowledge bases. Knowledge visualization aims to improve the creation, comprehension and transfer of knowledge by exploiting graphical and natural language processing means.

Graphical representation of knowledge was intensively studied in the previous decades and is still ongoing research (please refer [28] for a survey). The popularization in the use of Ontologies brings into focus the necessity to provide graphical visualization as an essential feature for every tool for ontology editing and browsing. Tree and graph visualization approaches are the more common techniques to graphically represent ontologies. A concrete solution for displaying large tree structures, called *hyperbolic tree* [29], was developed in 1995 in Xerox Parc Laboratories and commercialized by Inxight[5] . This technique is used in tools like KAON[6] and KIM[7].

[5] http://www.inxight.com
[6] http://kaon.semanticweb.org/
[7] http://dell.sirma.bg/kim/graph/Graph.jsp

A complementary approach for knowledge visualization in which semantic data descriptions are presented in a user friendly way is natural language generation (NLG). "*NLG takes structured data in a knowledge base as an input and produces Natural Language text, tailored to the presentational and the target reader*" [30]. NLG mechanisms can constantly keep up-to-date text descriptions of data semantics and can automatically provide those text descriptions in multiple languages [31]. Current efforts in NLG have two main focuses. The first one is to provide tools specific oriented to semantic web platforms, and the second one is to design NLG systems that keep the system simple enough to be maintained by non-NLG experts, but without losing quality of the text output ([32], [33], [34], and [35]). Since I expect that most of the users will not be NLG experts, these proposals can be very useful for the Semantic Web.

2.5 Distributed Architectures for storing and sharing data

The Web has been described using an abstract model called REST (Representational State Transfer) [36]. The fundamental principle of REST is that resources are stateless and identified by URIs. [37] demonstrates that it is not possible for a server to transmit any information to a client asynchronously in REST because every representation transfer must be initiated by the client, and every response must be generated as soon as possible (the statelessness requirement). Asynchronous communication is a requirement that CSpaces will require, so we have studied several extensions of REST, like ARRESTED [37].

Peer-to-Peer system is an interesting proposal for decentralized, distributed, self-organized systems, capable of adapting to changes such and failure [38]. Although there are several open issues regarding scalability, shared resources management, security and trust [39], current efforts in the field [40,41] are progressively overcoming these problems.

P-Grid[8], Edutella[9] and OceanStore[10] are current state of the art in P2P systems. They are examples of decentralized architecture that address several interesting problems like optimizing message flooding (HyperCup in Edutella), increasing fault tolerance using replication (OceanStore), improving security via cryptography techniques (OceanStore) and generation of public key in a decentralized manner (P-Grid). However, none of them provide all features that can be desirable, so implementation of extensions to one of these systems should be studied.

[8] http://www.p-grid.org/

[9] http://edutella.jxta.org/

[10] http://oceanstore.cs.berkeley.edu/

3. BUILDING BLOCKS TOWARDS CONCEPTUAL SPACES (CSPACES)

CSpaces are built around six building blocks. Given space limitation of this paper, I will introduce briefly all of them and in the following subsections will describe in more detail four of them.

Semantic data, schema and organizational model. Data elements and their relations should be described using formal representation languages (e.g. RDF and RDF(S) for the Semantic Web) that include a set of modeling primitives. The relations between data elements and data properties should be constructed as Ontologies. Ontologies are distributed in Individual and Shared CSpaces where heterogeneity should be overcome by finding a unified representation of the information in which mismatches are identified and transformation rules are implemented.

Coordination model. CSpaces is a middleware infrastructure for applications and a cooperation infrastructure for humans that aims to unify on the one hand, several communication means like e-mail, faxes, weblogs, etc, and on the other hand, coordination mechanisms like tuplespace, publish-subscribe, message queues, etc. The coordination model combines two metaphors: "persistent publish and read" and "publish and subscribe".

Semantic interoperability and consensus making model. The identification and representation of mismatches, and the definition of transformation rules will be required to ensure interoperability between the participants in different CSpaces. Moreover, consensus techniques are required to build Shared CSpaces. Humans need to reach an agreement about which information will be shared (content agreement) and how it will be represented (semantic agreement).

Security and Trust model. The protection of private and restricted information stored on spaces and the inclusion of trusted mechanisms to guarantee the validity, or trustworthiness of the information accessed are critical requirements for a successful development of a distributed information infrastructure.

Knowledge visualization model. Given the fact that the final goal of CSpaces is to minimize the amount of syntactic data representation (current Web) and maintain mostly semantic descriptions of data, it is necessary an infrastructure with knowledge visualization facilities to easily interpret the information stored through graphical and natural language generation mechanisms.

Architecture model: scalable, decentralized, distributed and secure are four design goals that CSpace infrastructure has to achieve. It would be also interesting the implementation of replication, searching, routing and data allocation (besides others) services.

3.1 Defining Conceptual Spaces: semantic data, schema and organizational model

Nowadays, there is a debate in the knowledge representation field about how ontologies, rules and alignment specifications should coexist. [45] argues that a set of ontologies and alignment specifications is more suitable for the Semantic Web because provides a loosely coupled configuration where updates in the mapped ontologies can be easily incorporated. On the other hand, a merge configuration of several ontologies can provide a more coherent, compact and consistent reasoning space and avoid the necessity to include reasoning mechanisms that have to deal with inconsistencies. Also the applicability in real scenarios of query rewriting to query mapped ontologies is limited in practice [27]. Based on previous experiences, I propose a new organization of the machine processable semantics that will be provided by the Semantic Web. This organization is articulated around Conceptual Spaces.

A **Conceptual Space** (CSpace) is a finite set of ontologies, their instances, and mapping and transformation rules (alignment specification). All these elements are represented using a common formal language that allows ontologies to be enriched with rules[11], and exhibit some degree of semantic autonomy[12].

Ontologies, which provide an unambiguous definition of the meaning of terminology/vocabulary used to describe a concrete domain, are used as a skeletal foundation for a Knowledge Base. An ontology can be formally defined as a 4-tuple < C, R, I, A > [45] where C is a set of concepts, R is a set of relations, I is a set of instances and A is a set of axioms. An ontology has to be specified in a formal logical language, i.e. a formalism with a well-defined semantics (OWL[13] and RDF(S)[14], are current relevant standards for Semantic Web).

[11] Although we have not considered at this moment representation formalisms to characterize uncertainty or vagueness, future work should evaluate and integrate these mechanisms.

[12] Semantic autonomy represents a particular perspective of the world of an individual or group of agents (humans or not). This semantic autonomy is represented by a set of schema specifications that organize and classify information according with an individual or shared interpretation.

[13] http://www.w3.org/TR/2004/REC-owl-ref-20040210/

[14] http://www.w3.org/TR/rdf-schema/

Rules are generally of the form of an implication between an antecedent (body) and consequent (head). The meaning of a rule can be informally described as: "*whenever (and however) the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold*"[46]. A rule has the form: *consequent* ← *antecedent*, where both are a conjunction of atoms, R(t1, ..., tn) composed by variables and/or constants. Rules are becoming very popular in the Semantic Web area because they enable one to enrich ontological specifications ("*build ontologies on top of rules*" [46]) and to build rules using the vocabulary defined by ontologies ("*build rules on top of ontologies*" [46]). Rules can also be very useful to improve query capabilities and provide mapping descriptions and data transformation specification for data integration.

Alignment specification, a set of mapping descriptions and transformation rules to handle heterogeneity between two or more ontologies. Mappings are typically expressed by some form of logic programming style rules, offering the expressivity of a powerful query language are a natural choice. However, the mapping language to be used does not provide any formal grounding, leaving open the choice of the appropriate semantics for the mappings; as a consequence, a formal semantics compatible with Description Logics and/or Logic Programming can be defined [27].

All logical statements have associated three identifiers. The first one is the identifier of the context where they were created (**id_context**). The second identifier is a unique id for the logical statement (**id_statement**, which can simplify reification, and make the code more compact, and the third one is a unique version id (**id_version**) that identifies each version of a logical statement.

In a CSpace, I can distinguish between raw and reasoning sub-space. A **raw** sub-space stores imported or local data, schemas, and alignment specifications (mapping and transformation rules) between these schemas. On the contrary, a **reasoning** sub-space provides a compact representation of an associate raw sub-space. The main goal of this compact representation is to maximize reasoning performances.

Ontologies, rules and alignments are maintained and updated on raw sub-spaces (one for each CSpace). The associated reasoning sub-spaces are periodically re-generated with the last version of the raw sub-space. In this regeneration (or initial generation) process, ontologies and rules are merged based on the alignment specification stored in the raw sub-space. A refinement and validation process identifies inconsistencies in the merged ontology and supports the user during the resolution of those inconsistencies. The resultant knowledge base is optimized in order to achieve better reasoning performances. Current proposals are studying the applicability of language weakening, knowledge compilation and approximate deduction in the context of the Semantic Web.

I distinguish two types of CSpaces, individual CSpaces and Shared CSpaces. An Individual CSpace is a formal representation of the perception that each individual (human or not) has about the Semantic Web (or a limited part of it). The machine processable semantics stored in an Individual CSpaces can be private (only the owner of the space can access it), restricted (a limited number of individual can access it) or public (the information can be accessed without restriction). The combination of restricted and public data can be used to create Shared Conceptual Spaces. Shared CSpaces are conceptual spaces shared by several participants that have reached an agreement on how to represent semantically common concepts. This requirement is fundamental to ensure interoperability between participants.

CSpaces can be viewed as leaves and shared spaces can be graphically considered as the branches and the trunk of a fictitious tree following a very similar organization proposed in CO4 (Collaborative construction of consensual knowledge bases) [8]. CO4 is an infrastructure enabling the collaborative construction of a knowledge base through the web. The main contribution of this approach is a proposal for organizing KBs in the same way that we proposed for CSpaces, as a tree structure. The leaves are called user KBs, and the intermediate nodes, group KBs. Each group knowledge base represents the knowledge consensual among its sons (called subscriber knowledge bases). When a subscriber wants to extend their group knowledge base, he/she submits a proposal with the modifications to the other subscribers. In response, users must answer by one of the following: accept when they consider that the knowledge must be integrated in the consensual knowledge base, reject when they do not, and challenge when they propose another change.

On the contrary of CO4 where modifications need to be approved by all subscribers, the updates proposed by the members of a Shared CSpace are automatically included, and versioning mechanisms are in charge to track changes and provide rollback features if one of the members disagrees with the included updates.

To join a Shared CSpaces and publish and retrieve data on it the new members should first complete a registration procedure in which one of the main tasks is to provide a semantic and alignment specification between the data that each new candidate want to share and the data that previous members have published beforehand.

3.2 Coordination model: "publish, read and subscribe"

Thanks to the Web, humans can *persistently publish and read* information at any time stored on servers spread around the World. The "*persistent publish and read*" metaphor have been also applied successfully as a simple coordination model for parallel computing [47], and more recently to Semantic Web Services [3]. Tuple-Space [47] is a coordination mechanism in which synchronization and communication between participants take place through the insertion and removal of tuples to/from a common shared space. Shared, persistent, associative, transactional secure and synchronous/asynchronous communication are main properties of Tuple Space.

However, "*persistent publish and read*" has the drawback that does provide flow decoupling from the client side. The interaction model provides time, space decoupling but not flow decoupling [12]. A user who is interested on an update version of a concrete web page has to check periodically for new contents although those contents are not already published. This restriction is a consequence of the REST (Representational State Transfer) [36] architecture style that characterizes the Web. To overcome this limitation, the "persistent publish and read" metaphor has been extended into "*persistent publish, read and subscribe*". The popularization of "weblogs[15]" (blogs or bloggings) together with the development of RSS (Rich Site Summary or Really Simple Syndication, http://www.rss-specifications.com/) bring a new form of interaction for web-users based on content subscription[16].

TupleSpace has the same limitation from the reader-side. An application which wants to read a concrete tuple has to run a process that periodically checks if the data is available. JavaSpaces[17] and TSpaces[18], concrete java implementations of TupleSpace, provide a simple notification mechanism to mitigate the problem. Thus Event-based technology can complement Tuple-Space with a sophisticated notification and subscription mechanism that allow a proper asynchronous interaction from the consumers/reader side. Additionally, in Tuple Space, it is not possible to know before hand which is the format of the data that producers will use to publish information in the space, and therefore, there is no way to know which data format the consumers expect. An implicit agreement is expected, but in the Semantic

[15] A website which stores miscellaneous notes updated daily and published in chronological order [18]

[16] RSS is still based on polling, but it is invisible for end-user. Users subscribe, but their RSS client is just polling the webserver every x minutes.

[17] http://java.sun.com/developer/products/jini/index.jsp

[18] http://www.research.ibm.com/journal/sj/373/wyckoff.html

Web where millions of users and applications will interact, these implicit agreements are not feasible.

On the other hand, a main disadvantage of most TupleSpace and Event-based implementations systems is the limited ability to define matching templates. Since CSpaces will store schema and data semantics, the coordination model based on the integration of TupleSpace and Event-based technologies have to be extended to support machine processable semantics[19]. For instance, in Triple Space Computing [3] the data published and accessed is represented by RDF triples. [48] proposes an equivalent approach for event-based systems using DAML+OIL to express a more accurately subscriptions and to improve event filtering mechanisms. CSpaces integrate tuplespace, event-based operations and semantic data specification in a new coordination model. Table 2 shows a reduced version of the first specification of the coordination model API for CSpaces. This coordination API is inspired by the combination of TSpace API and SIENA API[20].

Table 2. Coordination Model API for CSpaces

API call and description		
Void	write	(set tuples, IdCSpace id)
	Write one or more tuples in a concrete CSpace identified by a unique id. This method is shared by tuple-space and Event-based implementation	
Tuple	take	(Template t, IdCSpace id)
	Return a tuple (or nothing) that match with the template (that can be expressed using a formal query language) and delete the matched tuple from a concrete CSpace	
Tuple	waitToTake	(Template t, IdCSpace id)
	Like take but the process is blocked until the a tuple is retrieved	
Tuple	read	(Template t, IdCSpace id)
	Like take but the tuple is not removed	
Tuple	waitToRead	(Template t, IdCSpace id)
	Like read but the process is blocked until the a tuple is retrieved	
Set	scan	(Template t, IdCSpace id)
	Like read but returns all tuples that match with t	
Long	countN	(Template t, IdCSpace id)
	Return the number of tuples that match template t	
Void	subscribe	((IdSubscriber s, Template t), IdCSpace id)
	A subscriber expresses its interested on tuples that match with template t in a concrete CSpace. Any time that there is an update in the CSpace, the subscriber receives a notification that there are tuples available that match the template	
Void	unsubscribe	((IdSubscriber s, Template t), IdCSpace id)
	A subscriber deletes its subscription, and no more related notifications are received	
Void	advertise	((IdPublisher p, Template t), IdCSpace id)
	A publisher shows its intention to provide tuples that match t	
Void	unadvertise	((IdPublisher p, Template t), IdCSpace id)

[19] CSpace coordination model = "persistent publish, read and subscribe" + "semantics"
[20] http://serl.cs.colorado.edu/~carzanig/siena/

API call and description
A publisher shows will not provide more tuples that match t

In a Semantic Web mostly composed by machine processable semantics, "persistent publish, read and subscribe" metaphor can be the common interaction model for machines and humans. I think that this metaphor is flexible enough (as Weblogs, Tuple Space and Event-based demonstrated) to progressively substitute other communication means like fax, e-mail, instant messages, and present Web pages. By unifying those communication channels, CSpaces coordination model can contribute to mitigate one of the most relevant sources of information overload.

3.3 Knowledge Visualization

At this initial stage, Knowledge visualization user interface will be implemented reusing and combining two different approaches. I will provide a graphical navigation interface based on *TouchGraph*[21] (a popular general-purpose hyperbolic tree visualization library). To facilitate the understanding of the information showed by the graphical interface, I decided to integrate ONTOSUM [35], a generator for textual tailored summaries from ontologies. I chose ONTOSUM because is based on a well tested technology [32], it is domain-independent, it is designed for non-NLG experts, and it supports entries in different formal ontology languages like RDF(S), DAML+OIL and OWL.

ONTOSUM is implemented as a pipeline system [30] inside of the GATE infrastructure [49]. Althought the integration with GATE reports a lot of benefits, it would be interesting to disaggregate the NLG components and build an independent tool that can be executed in light-weight devices. The generator, HYLITE+, is implemented in Prolog and can run in diverse platforms. However, other components for preprocessing data semantic inputs are required (including a light-weight ontology called PROTON[22]), and thus, they should be adapted to work independently from the GATE system.

3.4 Architecture model: a decentralized hybrid approach

Given that CSpaces aims to re-elaborate the Semantic Web proposal by minimizing syntactic data representation, many of the design considerations

[21] http://sourceforge.net/projects/touchgraph/
[22] http://proton.semanticweb.org/

for the Semantic Web architecture are still valid for CSpaces. Scalable, distributed and decentralized are three requirements that CSpace and Semantic Web architectures have in common. However, the CSpace coordination model built over "persistent publish, read and subscribe" metaphor requires an architecture model that can deal with asynchronous communication. Furthermore, the organization of metadata around Individual and Shared CSpaces is another different in both infrastructures.

Like Semantic Web, my first idea was to build CSpaces upon existing Web infrastructure that has been described using an abstract model called REST (Representational State Transfer) [36]. The fundamental principle of REST is that resources are stateless and identified by URIs. HTTP is the protocol used to access to the resources and provides a minimal set of operations enough to model any applications domain [36]. Those operations (GET, DELETE, POST and PUT) can be easily mapped to Tuple-Space operations (READ, TAKE and WRITE in TSpaces). Tuples can be identified by URIs and/or can be modeled using RDF triples (as [3] suggests). 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. Furthermore, there is no direct way to model a peer-to-peer relationship [37]. Several extensions of REST, like ARRESTED [37], have been proposed to provide a proper support of decentralized and distributed asynchronous event-based web systems.

The limitations of REST to model asynchronous interaction motivated that I pay attention to Peer-to Peer [12] systems. They are decentralized, distributed, self-organized and capable of adapting to changes such as failure [38]. Although there are several open issues regarding scalability, shared resources management, security and trust [39], current efforts in the field [40,41] are progressively overcoming these problems.

My preliminary proposal for CSpaces architecture is strongly influenced by the work done in OceanStore[23], Edutella[24] and SWAP[25]. I distinguish between three kinds of nodes: CSpace-servers, CSpace-heavy-clients and CSpace-light-clients.

CSpace-servers store primary and secondary replicas of the data published in individual and shared CSpaces; support versioning services; provide an access point for CSpace clients to the peer network; include reasoning services for evaluating complex queries; implement subscription mechanisms related with the contents stored; balance workload and monitor

[23] http://oceanstore.cs.berkeley.edu/
[24] http://edutella.jxta.org/
[25] http://swap.semanticweb.org/public/index.htm

requests from other nodes and subscriptions and advertisements from publishers and consumers.

CSpace-heavy-clients provide a storage infrastructure and reasoning support to let users to work off-line with their own individual and shared spaces. Replication mechanisms are in charge to keep replicas in clients and servers up-to date. In addition, these clients also include a presentation service (based on NLG and Knowledge visualization techniques) to facilitate the visualization and edition of knowledge contents.

CSpace-light-clients only include the presentation infrastructure to query, edit and visualize knowledge contents stored on CSpace-servers.

When clients are online and connected with the rest of the nodes of the system through an access point (server node) they have the obligation to share computational resources (CPU time, memory and persistent storage services). Thus CSpace-servers can divert to client's resources demanding requests, and consequently, alleviate temporary the workload of servers. If the client is a heavy-client, requests that can be performed locally will not be sent to CSpace-servers. Periodically, replicas will be updated to keep heavy-clients and servers up-to-date.

I call these types of systems hybrid because elements of both pure P2P and client/server systems coexist. CSpace-servers are formally peers, but it is not the case of CSpace-clients that promote a client-server relation with CSpace-servers. Like OceanStore [40], this configuration drives into two-tiered system. The upper-tier is composed by well-connected and powerful servers, and the lower-tier, in contrast, consists in clients with limited computational resources temporary available.

It is expected that CSpaces infrastructure will be self-organized like in other peer-to-peer systems and will include monitoring mechanisms that will analyze the distribution of the data in the different nodes and the data flows between these nodes. Servers and clients will be re-distributed in appropriate configurations that minimize the network traffic and maximize semantic similarity of the data stored in closer peer. Subscriptions and advertisements from publishers and consumers will provide useful information to determine optimal configurations where consumers and publishers with common interests will be connected to closer servers. In addition, the definition of Shared CSpaces will be other information source to determine semantic similarity between nodes.

Communication metaphor will differ from most of the P2P implementation that use message passing. Like OceanStore is built on top of an event-based architecture[26], CSpace promotes the coordination model "publish, read and subscribe" for the communication of its nodes. In

[26] More precisely is Pond [22], the OceanStore prototype, which is built on top of an event-based system

addition, the use of topologies that simulate spanning trees (i.e. HyperCup in Edutella) will reduce unnecessary data flows and will facilitate the implementation of replication mechanisms.

Given the decentralized nature of CSpace infrastructure, solutions for security and trust mechanisms have to rely on the same principles. Currently, there are several proposals that aim to provide in a decentralized manner solutions for building public key infrastructures [42], restricting access to concrete contents [40], avoiding manipulation from malicious peers of data flowing in the network [43], and defining trust values for each peer without centralized globally-trust servers [44]

OceanStore incorporates several interesting solutions that will be studied in detail to verify their suitability for CSpaces. The authors of OceanStore propose a distributed system where read-only versions of the data will be kept it in the system. Support for fault tolerance is achieved by including a configuration of primary and secondary replicas, and the use of a technique that splits each data object into several fragments that can be dispersed in different servers. Those fragments can be recovered to re-construct the original data object even if some of the fragments are not accessible. Cryptography features are also incorporated to increase security access to the system and mechanisms to guarantee the validity, or trustworthiness of the information accessed.

4. SPACES IN THE REAL WORLD

In my opinion, the combination of an intensive use of semantic data descriptions, a flexible coordination model and the implementation of visualization mechanisms based on graphical and natural language technologies can be the basement for a new set of applications for industry, research and education areas. In particular, I am investigating the applicability of CSpaces in personal and distributed knowledge management, enterprise application integration, Semantic Web Services, software components coordination and ubiquitous computing. The former is closely related with the problem of information overload. In the following subsections, it will be briefly introduced these potential application scenarios for CSpaces.

4.1 Personal and distributed knowledge management

*"**Personal Knowledge Management (PKM)** is a collection of processes that an individual needs to carryout in order to gather, classify, store, search and retrieve knowledge in his/her daily activities. Activities are not confined*

to business/work-related tasks but also include personal interests, hobbies, home, family and leisure activities" [61]. As today's knowledge workers often have to deal with data, information and knowledge specified in various formats (e.g. hard copy, video, picture, texts, voice message etc.), distributed using different information channels (e.g. emails, fax, instant messages, file systems, etc) and stored using multiple electronic devices for communications, planning and recording purposes [61].

A potential way to alleviate this situation is to use semantic data representation of the information that it is received or distributed, and to reduce the number of information channels. As it was mentioned earlier in the paper, I envision "persistent publish, read and subscribe" metaphor for machine processable semantics as the common interaction model for machines and humans that progressively substitute other communication means. By unifying those communication channels, CSpaces coordination model can contribute to mitigate one of the most relevant sources of information overload and to improve personal knowledge management. Furthermore, the use of knowledge visualization and natural language processing techniques will facilitate the manipulation of data semantics by humans and consequently will reduce significantly the amount of syntactic data representation and textual information. The benefit of this reduction will minimize duplicities (syntactic and semantic) of data representation.

CSpaces can contribute to organize and share knowledge using a bottom-up approach. Instead of centralized systems that forces users to agree in a set of rules, schemas and data, CSpaces offer a distributed infrastructure where users can publish personal knowledge that can be shared with other users with common information/interests. This approach is inspired in an earlier proposal called Distributed Knowledge Management [6, 7] where its authors confirmed during the realization of several tests in real scenarios that users were more favorable to this kind of approach because it takes into account the different perspectives and understandings that users have about the world and more concretely about the information, processes and interactions of their organizations or working groups. The combination of Individual CSpaces can generate a new space shared by all these users. Shared CSpaces is built on top of a semantic data representation agreement of a group of users. Moreover, Shared CSpaces can be combined to generate bigger Shared CSpaces, or in other words, bigger knowledge repositories.

4.2 Enterprise Application Integration

Given that one of the main goals of CSpaces is to provide a collection of homogeneous semantic spaces in which heterogeneity of data sources are reconciled, CSpaces can be an excellent approach to handled smoothly Enterprise Application Integration (EAI)[27].

Integration and heterogeneity are two concepts that come together. Heterogeneity is one of the hardest problems that humans and machines have to overcome in order to ensure interoperability, and in a distributed and open system like Internet, heterogeneity cannot be avoided [50]. Several attempts to classify source/levels/categories of heterogeneity can be found in the literature (refer [50] for a survey).

The process of reconciling differences between heterogeneous information sources is called mediation [51], and in the particular case of Ontologies is called Ontology mediation [45]. In general, the identification and alignment of heterogeneity between several ontologies or data sources in a mediation process is not fully automatic and in case of complex data sources very time consuming.

Many frameworks assumed wrongly that mediation can be done "on the fly". In my opinion, the initial identification and alignment of heterogeneity in a mediation process should be done before any possible interaction or collaboration, and because usually there are several ways to reconciliate two or more heterogeneous data sources, it is expected an interactive mechanism that facilitate consensual decisions between involved actors. Shared CSpace is this consensual mediated space where applications can publish and share their data. During the registration process to a Shared Space, applications publish relevant ontologies and rules using a consensual specification. Through the publish-subscription mechanism, applications indicate which kind data will publish in the space and which kind of data would like to consume. Thanks to the CSpace coordination model applications interact through the space by simple means of publishing and reading data semantically described.

4.3 Towards Semantic Event Oriented Architecture for Web Services (SEOA-WS)

The case of enterprise application integration can be easily extrapolated to Web Services and software components coordination. Web Services are built over three main building blocks: service oriented architecture, redesign of middleware protocols and standardization [10]. *Service Oriented Architecture* (SOA) is based on the idea that companies will publish interfaces of their applications as services that can be invoked by clients. The second block, the *redesign of middleware protocols* to work in decentralized

[27] EAI is the unrestricted sharing of data and business processes throughout the networked applications or data sources in an organization (http://www.webopedia.com/TERM/E/EAI.html).

environment in order to overcome the limitations of centralized middleware architectures in terms of trust and confidentiality. Finally, the last key block is a set of *standard languages* and protocols that eliminates the necessity of many different middleware infrastructures. As a relevant part of the Service Oriented Architecture (SOA), notification is expected to play an essential role in the development of asynchronous, loosely-coupled and dynamic systems, where entities receive messages based on their registered interest in certain occurrences or situations. WS-Notification [52] and WS-Eventing [53], bring again the event-based communication paradigm to the fore.

Following the main principles that the Semantic Web introduced to extend the current Web, Semantic Web Services proposes to add machine processable semantics to Web Services in order to reduce manual efforts during the deployment of integration of distributed applications by improving automation in the location, combination and use of Web Services. For software architects, Semantic Web Services are the building blocks to evolve Service Oriented Architecture into **Semantic Service Oriented Architecture** (**SSOA**). Unfortunately "*adding semantics*" is not enough. Heterogeneity has become in an insurmountable obstacle that current proposals for web services and semantic web services[28] are not able yet to tackle. The vision of a global distributing computing through web services only could become true if all the participants involved provide mechanisms to achieve a common explicit formal understanding of their semantic specifications.

In my opinion, Shared CSpaces can facilitate a more effective discovery, invocation and interoperation of Semantic Web Services that are register to the same space. Heterogeneity problems are reduced thanks to the mediated data semantics published in Shared CSpaces and the coordination model of CSpaces provides a simple interaction mechanism for Semantic Web Services. The description of the data that the semantic web services registered in a Shared CSpace plan to publish is stored in the publisher register, and the data that those services plan to consume is stored in the subscriber register. These descriptions of data publication and consumption can be viewed as a very simplified version of services capabilities and goals [54]. Moreover, I am currently studying the practicability of evolving WSMX architecture[29] into Semantic Event Oriented Architecture for Web Services (SEOA-WS), SSOA architecture based on event mechanisms. I am

[28] The inclusion of Mediators as a part of the WSMO (http://www.wsmo.org) architecture is a promising initiative to confront the problem of heterogeneity.

[29] WSMX (http://www.wsmo.org/wsmx/) is a reference implementation of an execution environment for the dynamic discovery, selection, mediation, invocation and interoperation for Semantic Web Services based on WSMO specification. WSMX follows SOA principles.

working in the integration of a simplify version of CSpace as a data semantic repository and coordination model for the components of the system [15].

4.4 Ubiquitous Computing

Pervasive or *Ubiquitous computing* was the vision of Mark Weiser[30] for a World saturated with computing and wireless communication gracefully integrated with human users. One of the typical examples of ubiquitous computing applications is *active environments* ([55, 56, 57] and [58]). Active environments are sensor-rich environments with computational and communication facilities that analyze users behave to anticipate potential new requirements or facilitate the developing of certain tasks in a natural way. The large number of sensors required in this kind of environment potentially produces a vast amount of data that it is necessary to filter to potential consumers to allow more efficient performances, and avoid their saturation. Thus, a middleware infrastructure is required to cope with high-volume data and be able to aggregate and transform it before dissemination.

Started at Stanford University in mid-1999, the Interactive Workspaces[31] project is a concrete example of application of the idea of active environment in laboratories and collaborative e-learning spaces[32]. One of the versions of the prototype for interactive workspace, called the iRoom [58], included iROS (Interactive Room Operating System), a middleware infrastructure designed to be embedded in all devices that belong to the iRoom. One of the components of this software is Event Heap [59], a coordination mechanism derived from a tuple space model, and implemented on top of the TSpaces (Tuple Spaces) system from IBM Research [60]. Unlike TSpaces, the Event Heap treats the fields as an unordered collection and allows references to tuple fields only by name, not by index. In addition, applications can specify incomplete query templates with only some combination of name, type and value for fields.

CSpaces can improve the Event Heap mechanism with a publish-subscription mechanism embedded in the Tuple Space system that provide a asynchronous behave from the client side and can facilitate the identification of data requirements from producers and consumers by analyzing the subscriptions stored in the system. Moreover, the use of data semantics would allow a richer specification of the information that flows through the system and would improve searching capabilities for retrieving operations.

[30] http://www2.parc.com/csl/members/weiser/
[31] http://iwork.stanford.edu/main.shtml
[32] http://www.stanford.edu/dept/SUL/acomp/teamspace/

5. CONCLUSIONS

CSpaces aim to create an infrastructure that reduces the problem of information overload and facilitates collaboration between machines, humans, and humans-and-machines, based on an intensive use of machine processable semantics. *"Publish, read and subscribe"* of machine processable semantics, a re-elaboration of Tuple Space and Publish-Subscribe systems, is the communication mean that pretend to reduce heterogeneity produced by a diverse set of heterogeneous information channels (emails, fax, instant messages, web pages, etc). Through natural language and knowledge visualization techniques, humans will be able to interact with this "purest semantic" web.

Machine processable semantics will be published and shared in CSpaces, a finite set of instances, ontologies, and mapping-and-transformation rules (alignment specification) that are represented using a common formal language, and exhibit some degree of semantic autonomy. To improve reasoning performances and do not difficult the update of the data stored, CSpaces maintain a reasoning sub-space and a raw sub-space, respectively.

Individual and Shared CSpaces will provide a logical organization, inspired in a tree model, of machine processable semantics in the Semantic Web. Individual CSpace is a formal representation of the perception that each individual (human or not) has about the Semantic Web (or a limited part of it). On the other hand, Shared CSpace represents the agreement of a group of humans and/or machines of how formally represent concepts of their Individual CSpaces. Through "publish, read and subscribe", humans and/or machines will be able to interoperate using a common Shared CSpace. Access rights associated to Individual and Shared CSpaces will assure different levels of privacy of the information published.

To complete the presentation of CSpaces, I sketched in section 3 the architecture that can take into account the requirements imposed by the Semantic Web and the "publish, read and subscribe" coordination paradigm. Since REST cannot model this coordination paradigm, and hybrid architecture based on P2P is my choice for CSpace infrastructure. Three kind of nodes (CSpace-servers, CSpace-heavy-clients, and CSpace-light-clients) are defined, and several proposals for security, trust and P2P topologies and communication mechanisms are discussed.

The paper concludes with a briefly description of the applicability of CSpaces in four different scenarios: personal and distributed knowledge management, Enterprise Application Integration (EAI), Semantic Event oriented Architecture for Web Services (SEOA-WS), and Ubiquotous computing.

CSpaces is in an early stage, so as a future work, I will continue the process of refining the ideas presenting in this paper, testing new tools that can be used in this framework, and working in the implementation of a first prototype.

ACKNOWLEDGEMENTS

I would like to thank Ying Ding and Uwe Keller (Leopold-Franzens Universität Innsbruck, Austria), Stefan Decker, Mathew Moran and Eyal Oren (National University Ireland Galway) and Manfred Hauswirth (Ecole Polytechnique Federal de Lausanne, EPFL) for fruitful discussion and feedback.

This work is funded by the European Commission under the projects Knowledge Web; by the Vienna city government under the CoOperate programme and by the FIT-IT (Forschung, Innovation, Technologie - Informationstechnologie) under the project TSC.

REFERENCES

1. School of Information Management and Systems (SIMS). How much Info? (Technical Report). University of Berkeley. 2003.
2. T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web. Scientific American, May 2001.
3. D. Fensel: Triple-based Computing. Digital Enterprise Research Institute (DERI) Technical Report DERI-TR-2004-05-31. 2004
4. J. de Bruijn, F. Martin-Recuerda, D. Manov, M. Ehrig, 4.2.1, State-of-the-art survey on Ontology Merging and Aligning V1, SEKT, 2004
5. K. Kotis, G.A. Vouros, J.P. Alonso, HCONE: tool-supported methodology for collaboratively devising living ontologies. In SWSDB 2004: 2nd int. Workshop on Semantic Web Databases , 2004
6. 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
7. M. Bonifacio, P. Bouquet, G. Mameli and M. Nori, Peer-Mediated Distributed Knowledge Management, 2003, http://eprints.biblio.unitn.it/archive/00000426/01/032.pdf
8. J. Euzenat, Building consensual knowledge bases: context and architecture, in N. Mars (ed.), Towards very large knowledge bases, IOS press, Amsterdam (NL), pp143-155, 1995
9. A.D. Birrell and B.J. Nelson, Implementing remote procedure calls. ACM Transactions on Computer Science, 2(1):39-59, Feb. 1984
10. G. Alonso, F. Casati, H. Kuno, and V. Machiraju (2004). Web Services. Springer, 2004.
11. D. Gelernter, N. Carriero, S. Chang: Parallel Programming in Linda, Proceedings of the International Conference on Parallel Processing. ICPP. 255-263. 1985

12. P. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. Technical Report. 2001.
13. C. Bussler, DERI-TR-2005-04-22 A Minimal Triple Space Computing Architecture, April 2005
14. Robert Tolksdorf, Lyndon Nixon, Franziska Liebsch, Duc Minh Nguyen and Elena Paslaru Bontas. Semantic Web Spaces (Technical Report, B 04-11). 2004. http://www.inf.fu-berlin.de/inst/pubs/tr-b-04-11.abstract.html
15. F. Martin-Recuerda and B. Sapkota, D21.v0.1 WSMX Triple-Space Computing, 2005, http://www.wsmo.org/TR/d25/d25.1/v0.1/
16. Andrea Cali, Giuseppe De Giacomo, Maurizio Lenzerini: Models for information integration: turning local-as-view into global-as-view in Proc. of the Workshop on Foundation of Models for Information Integration (FMII 2001): 10th Workshop in the series "Fondation of Languages, Data, and Object", LNCS, Springer.
17. S.S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J.D. Ullman, and J. Widom. The TSIMMIS project: Integration of heterogeneous information sources. In Proceedings of the 10th Meeting of the Information Processing Society of Japan, pages 7-18, October 1994.
18. M. Templeton, H.Henley, E.Maros, and D.J. Van Buer. InterViso: Dealing with the complexity of federated database access. The VLDB Journal, 4(2):287-317, 1995.
19. M.T. Roth and P. Schwarz. Don't scrap it, wrap it! A wrapper architecture for data sources. In Proc. VLDB'97, pages 266- 275, Athens, Greece, 1997.
20. A. Levy, A. Rajamaran, and J.Ordille, Querying heterogeneous information sources using source description. In Proc. VLDB'96, pages 252-262, 1996.
21. I. Manolescu, D. Florescu, and D. Kossmann. Answering XML queries on heterogeneous data sources. In Proc. VLDB'01, pages 241-250, 2001.
22. Jeffrey D. Ullman. Information integration using logical views. In Proc. of the 6th Int. Conference on Database Theory (ICDT 97), volume 1186 of Lecture Notes in Computer Science, pages 19-40. Springer Verlag, 1997.
23. S. Abiteboul and O. Duschka, "Complexity of Answering Queries Using Materialized Views" In Proc. 17th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 98), pp. 254-265, 1998.
24. P.J. McBrien and A. Poulovassilis. Schema Evolution in Heterogeneous Database Architectures, A Schema Transformation Approach. In Proceedings of CAiSE02, Springer Verlag LNCS, Volume 2348, Pages 484-499, 2002
25. Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini A framework for ontology integration. In Isabel Cruz, Stefan Decker, Jérome Euzenat, and Deborah McGuinness, editors, The Emerging Semantic Web _ Selected Papers from the First Semantic Web Working Symposium, pages 201-214. IOS Press, 2002.
26. L. Zamboulis and A. Poulovassilis. Information Sharing for the Semantic Web - a Schema Transformation Approach (Tech Report). 2005. http://www.doc.ic.ac.uk/automed/techreports/information_sharing_for_semantic_web.pdf
27. L. Predoiu, F. Martín-Recuerda, C. Feier, A. Polleres, F. Porto, A. Mocan, K. Zimmermann, and J. de Bruijn. Deliverable D1.5, DIP, 2004.
28. M.J. Eppler and R.A. Burkard. Knowledge Visualization. Towards a New Discipline and its Fields of Application, ICA Working Paper #2/2004, University of Lugano, Lugano.2004

29. Lamping, R. Ramana Rao and P. Pirolli, A Focus+context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. Conference on Human Factors in Computing Systems archive. In Proceedings of the SIGCHI conference on Human factors in computing systems. Denver, Colorado, United States. 401 - 408. 1995. ISBN: 0-201-84705-1
30. Ehud Reiter and Robert Dale. Building Natural Language Generation Systems. Cambridge University Press, Cambridge, England, 2000.
31. K. Bontcheva D5.1.1 Natural Language Generation for Knowledge Access. 2004
32. K. Bontcheva and Y. Wilks. Automatic Report Generation from Ontologies: the MIAKT approach. In Nineth International Conference on Applications of Natural Language to Information Systems (NLDB'2004), 2004.
33. G.Wilcock. Talking OWLs: Towards an Ontology Verbalizer. In Human LanguageTechnology for the SemanticWeb andWeb Services, ISWC'03, pages 109–112, Sanibel Island, Florida, 2003.
34. G. Wilcock and K. Jokinen. Generating Responses and Explanations from RDF/XML and DAML+OIL. In Knowledge and Reasoning in Practical Dialogue Systems, IJCAI-2003, pages 58–63, Acapulco, 2003.
35. K. Bontcheva. Generating Tailored Textual Summaries from Ontologies. Second European Semantic Web Conference (ESWC'2005). Crete. 2005.
36. R. T. Fielding. Architectural styles and the design of network-based software architectures. PhD Thesis, University of California, Irvine, 2000.
37. Rohit Khare and Richard N. Taylor. "Extending the Representational State Transfer (REST) Architectural Style for Decentralized Systems." Proceedings of the International Conference on Software Engineering (ICSE), May, 2004, Edinburgh, Scotland.
38. P. R. Pietzuch. "Hermes: A Scalable Event-Based Middleware". Ph.D. Thesis, Computer Laboratory, Queens' College, University of Cambridge, February 2004.
39. M. Bawa, B. F. Cooper, A. Crespo, N. Daswani, P. Ganesan, H. Garcia-Molina, S. Kamvar, S. Marti, M. Schlosser, Q. Sun, P. Vinograd, B. Yang. Peer-to-Peer Research at Stanford, The Peers Group. In SIGMOD Record, September 2003.
40. S. Rhea, P. Eaton, D. Geels, H. Weatherspoon, B. Zhao, and J. Kubiatowicz. Pond: the OceanStore Prototype. Appears in Proceedings of the 2nd USENIX Conference on File and Storage Technologies (FAST '03), March 2003
41. K. Aberer, P. Cudré-Mauroux, A.Datta, Z. Despotovic, M. Hauswirth, Ma. Punceva, R. Schmidt, P-Grid: A Self-organizing Structured P2P System SIGMOD Record, 32(2), September 2003.
42. K. Aberer, A. Datta, M. Hauswirth. A decentralized public key infrastructure for customer-to-customer e-commerce, to be published in International Journal of Business Process Integration and Management, 1(1), 2005.
43. B. Cooper, M. Bawa, N. Daswani, and H. Garcia-Molina. Protecting the PIPE from malicious peers. In FGCS 2004.
44. P. Sepandar D. Kamvar, Mario T. Schlosser and Hector Garcia-Molina.The EigenTrust Algorithm for Reputation Management in P2P Networks, In WWW, 2003.
45. J. de Bruijn and A. Polleres. Towards and ontology mapping language for the semantic web. Technical Report DERI-2004-06-30, DERI, June 2004.

46. B. N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: Combining logic programs with description logic. In Proc. of the Twelfth International World Wide Web Conference (WWW 2003), pages 48-57. ACM, 2003.
47. D Gelernter, N Carriero. Generative Communication in Linda. ACM Transactions on Programming Languages and Systems, 1985
48. H. Li and G. Jiang (2004) Semantic message oriented middleware for publish/subscribe networks. Proceedings of the SPIE, Volume 5403, pp. 124-133 (2004).
49. K. Bontcheva, A. Kiryakov, H. Cunningham, B. Popov, and M. Dimitrov. Semantic web enabled, open source language technology. In EACL workshop on Language Technology and the Semantic Web: NLP and XML, Budapest, Hungary, 2003.
50. P. Bouquet, J. Euzenat, E. Franconi, L. Serafini, G. Stamou and S. Tessaris. D2.2.1 Specification of a common framework for characterizing alignment. Knowledge Web Technical Report. 2004.
51. G. Wiederhold. Mediators in the Architecture of Future Information Systems, IEEE Computer, 25(3): 38-49, 1992.
52. S. Graham and P.Niblett (editors). Web Services Notification (WS-Notification). Technical Specification. 2004. http://www-106.ibm.com/developerworks/library/specification/ws-notification/
53. A. Geller (editor). Web Services Eventing (WS-Eventing). Technical Specification. August 2004. http://www-106.ibm.com/developerworks/webservices/library/specification/ws-eventing/
54. D. Roman, H. Lausen, U. Keller, D2v1.2. Web Service Modeling Ontology (WSMO), 2005, http://www.wsmo.org/TR/d2/v1.2/
55. M.C. Mozer. An intelligent environment must be adaptive . IEEE Intelligent Systems and their Applications, No. 2, Vol. 14, 1999, pp 11-3.
56. M.H. Coen. Building Brains for Rooms: Designing Distributed Software Agents. In Proceedings of AAAI/IAAI, 1997, pp 971-7.
57. B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, EasyLiving: Technologies for Intelligent Environments, Handheld and Ubiquitous Computing, September 2000
58. A. Fox, B. Johanson, P. Hanrahan and T. Winograd. Integrating Information Appliances into an Interactive Workspace. IEEE Computer Graphics & Applications, No. 3, Vol. 20, 2000.
59. Johanson, B. and A. Fox. The Event Heap: A Coordination Infrastructure for Interactive Workspaces. In Proceedings of the 4th IEEE Workshop on Mobile Computer Systems and Applications (WMCSA-2002). 2002. Callicoon, New York, USA.
60. P. Wyckoff, S. W. McLaughry, T. J. Lehman and D. A. Ford. TSpaces. IBM Systems Journal 37(3), 1998. Also available at http://www.almaden.ibm.com/cs/TSpaces
61. E. Tsui. Technologies for Personal and Peer-to-Peer (P2P) Knowledge Management. 2002. http://www.csc.com/aboutus/lef/mds67_off/uploads/P2P_KM.pdf

USING UDDI FOR PUBLISHING METADATA OF THE SEMANTIC WEB

Anton Naumenko, Sergiy Nikitin, Vagan Terziyan, Jari Veijalainen*
Industrial Ontologies Group, Department of Mathematical Information Technology, University of Jyvaskyla, FINLAND, e-mail: annaumen@cc.jyu.fi

* *Information Technology Research Institute, Faculty of Information Technology, University of Jyvaskyla, FINLAND, e-mail: veijalai@cc.jyu.fi*

Abstract: Although UDDI does not provide support for semantic search, retrieval and storage, it is already accepted as an industrial standard and a huge number of services already store their service specifications in UDDI. Objective of this paper is to analyze possibilities and ways to use UDDI registry to allow utilization of meta-data encoded according to Semantic Web standards for semantic-based description, discovery and integration of web resources in the context of needs of two research projects: "Adaptive Services Grid" and "SmartResource". We present an approach of mapping RDFS upper concepts to UDDI data model using tModel structure, which makes possible to store semantically annotated resources internally in UDDI. We consider UDDI as an enabling specification for creation of a semantic registry for not only services, but also for web resources in general.

Keywords: Web-Services, UDDI, Semantic Web

1. INTRODUCTION

Objective of the paper is to analyze possibilities and ways to use UDDI [UDDI] registry to allow utilization of meta-data, encoded according to Semantic Web [SemanticWeb] standards, for semantic-based description, discovery and integration of web resources in a context of needs of two research projects: "Adaptive Services Grid" (ASG) [ASG] and "SmartResource" [SmartResource], [Kaikova2004].

According to a definition by Moreau *et al* [Moreau2005] Semantic Discovery is the process of discovering services capable of meaningful interactions, even though the languages or structures, with which they are described, may be different. In the paper authors evaluate existing approaches, basically, UDDI with keywords-based search, describe a solution to extend service descriptions using RDF [RDF] and changes to UDDI APIs needed to support a semantic search.

A description of entities using Semantic Web standards is called a semantic annotation or simply an annotation. The annotation of an entity is a prerequisite to allow semantic discovering and integration. In the context of UDDI, an entity of the semantic annotation is usually a Web Service and more rarely businesses, business services and technical information that is a target of a binding. We go beyond trying to consider UDDI as a basis to create a semantics-enabled registry of resources from point of view of a Semantic Web domain. Moreover, we consider each resource entity (not just a web service) as a subject of the semantic annotation, registering, discovering, composition, enactment, integration, etc.

Different attempts to bring semantics to UDDI were faced in a number of papers. They consider mainly the process of publishing semantic information to UDDI registry with or without changes to existing UDDI APIs and data model. Additionally, they focus on the process of a semantic search based on an internal enhanced matchmaker with changes to UDDI APIs or an external matchmaking engines through creating a proxy API above UDDI, matchmaker and ontology.

UDDI+ server [Pokraev2003] is a good example of a solution when UDDI is used unchanged, but inside architecture of the server, which introduces additional elements like a matchmaker, an ontology repository and a proxy API to invoke UDDI APIs. Such the solution requires mapping a semantic language, in this case DAML-S [DAML-S], to UDDI publish message while keeping standard UDDI Publish and Inquiry interface.

Nowadays, some research efforts are focusing on experiments with commercial UDDI registries [Kawamura2003], [Kawamura2004], [Paolucci1], [Paolucci2] trying to provide a semantic search based on an externally created and operated matchmaker. Web Service Semantic Profile (WSSP) serves as the semantic annotation of a service and extends WSDL [WSDL] description of the service using RDF, RDFS [RDFS], DAML+OIL [DAML] or OWL [OWL], RDF-RuleML [RuleML]. Semantic data are stored outside of UDDI while keeping a link from corresponding tModel of a Web Service registered with UDDI to its WSSP.

Srinivasan *et al* [Srininvasan2004] provides a research close to target of this paper and describes a mapping of an OWL-S profile to the UDDI data model for a matchmaker architecture based on the Paolucci's results [Paolucci1]. The difference from our approach is that the OWL-S to UDDI mapping is done on a conceptual level while in this paper we try to map an underlying RDF model to a structure of tModel.

The remaining content is organized as follows. Chapter 2 summarizes needs of the ASG project to create a service registry to store semantic descriptions of Web- and Grid Services in addition to other semantically encoded information like rules, facts, domain knowledge, etc. Chapter 3 provides a description of a business use case of the SmartResource project and needs of a registry for semantically annotated resources. Chapter 4 shortly presents architecture and information model of UDDI. Chapter 5 describes an approach to bring semantics to UDDI by encoding RDFS upper concepts using the data model of UDDI tModel. Chapter 6 concludes the results of the analysis performed.

2. SEMANTIC WEB AND RESOURCE DESCRIPTION FRAMEWORK

The Semantic Web is an idea of World Wide Web inventor Tim Berners-Lee that the Web as a whole can be made more intelligent and perhaps even intuitive about how to serve user's needs. Nowadays Semantic Web Activity has produced several standards for a specification of arbitrary domain knowledge with a rich semantic expressiveness.

Semantic Web is expected to become a next-generation of the Web assuming that besides an existing content there will be a conceptual layer of machine-understandable metadata, making the content available for processing by intelligent software. This allows automatic resource integration and provides interoperability between heterogeneous systems. The next generation of intelligent applications will be capable to make use of such metadata to perform resource discovery and integration based on its semantics. Semantic Web aims at developing a global environment on top of Web with interoperable heterogeneous organizations, applications, agents, web services, data repositories, humans, and so on. On the technology side, Web-oriented languages and technologies are being developed (e.g. RDF [RDF], OWL [OWL], OWL-S [OWLS], WSMO [WSMO], etc.), and the success of the Semantic Web will depend on a widespread industrial adoption of these technologies. A trend within worldwide activities related to Semantic Web definitely shows that the technology has emerging grows of an interest both academic and industry during a relatively small time interval. The growing interest to the Semantic Web, as a research and educational domain, from the academy is evident. New scientific results and interesting challenges in the area appear rapidly. International networks cover topics related to intersections of various former scientific domains with Semantic Web technology and discover new challenging opportunities. Basic standards have been announced and the amount of pilot tools and applications around these standards is exponentially increasing. In spite of the growing hype around Semantic Web and appropriate standards, industry developed and is continuously developing own standards for interoperability and integration.

The Resource Description Framework (RDF) is a framework for representing information in the Web [RDF]. It is intended for integration of a variety of applications using XML [XML] for syntax and URIs for naming [SemanticWeb]. The RDF is a structure for describing and interchanging metadata on the Web [Powers2003]. The RDF is expressive and flexible technology to describe arbitrary domains and thus it is widely applicable. The World Wide Web Consortium (W3C) has been designing RDF as a basis technology to support Semantic Web Activity and it gives the following statement to describe the RDF: *The RDF is a language designed to support the Semantic Web, in much the same way that HTML is the language that helped initiate the original Web.*

The RDF is a framework for supporting resource description, or metadata (data about data), for the Web. RDF provides common structures that can be used for interoperable XML data exchange [SemanticWeb]. The RDF gives tools to developers to encode meaning by expressing concepts of problem domain and relations between them using RDF statements and connecting these statements to a semantic network. RDF, like XML and relational databases, follows object-based domain decomposition for data representation, but remains more generic and more expressive. There are also variety of software tools to work with RDF including tools for creating RDF, for creating vocabulary for RDF called Schema (RDFS), for querying RDF, for making inference based on an RDF defined semantic network, etc.

RDF brings to XML technology the same functionality as relational algebra to commercial database systems. RDF defines classes of problem domain concepts and their properties to create a vocabulary of the domain in the same way like a creation of tables and relationships between tables defines a schema of a database. XML can encode contents of a relational database and can encode the contents of an RDF-based model – but XML is not a replacement because XML is nothing more than syntax. A metadata vocabulary is needed to be able to use XML to record business domain information in such a way that any business can be documented, and RDF provides this capability [Powers2003].

3. ASG APPROACH TO DESCRIPTION, DISCOVERY AND INTEGRATION OF SERVICES

The main objective of the ASG-project is to develop an open generic software platform for adaptive services discovery, creation, composition and enactment. The ASG-platform is divided into several components having separate roles and interacting with each other by means of interfaces. The component structure is presented in Figure 1.

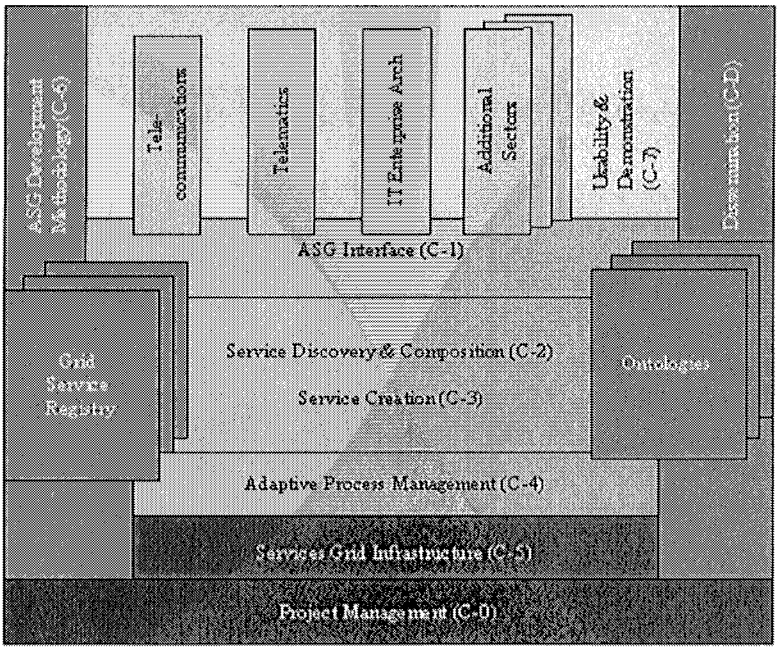


Figure 1. Component Structure (adopted from [ASG])

The key element of the ASG-platform is a persistent storage for platform data, so-called Registry. To understand a role of Registry, first let us consider business use cases.

In the following use cases a role of the Registry is quite straightforward. Registry represents the centralized platform storage containing various data about existing services and ontology specifications for domains involved in the service execution. A scenario used in the following use cases is a Traveling Tourist Scenario. It has following assumptions:

- User goal
  - Plan travel route from a departure location to a specific destination
- User specifies travel parameters:

    - Stopovers (e.g. city), preferred travel means, max. cost, travel duration, points of interests, etc.,
- Possibly needed services:
    - Find nearby Point of Interests (POI)
    - Find nearby Hotels
    - Receive several alternative Travel Routes for including stay (Hotels) and sightseeing (POI)

To achieve the desired user's goal, the ASG-platform provides a workflow which defines the sequences and an order of the sequence peers invocation. The inputs and outputs of services are adjusted using grounding to a common ASG-ontology.

The use-cases of the ASG-project constitute quite disperse requirements to Registry. Registry must store atomic service specifications, composed of a service name and a specification of its inputs, outputs, pre- and post-conditions. As far as all parameters refer to the common platform and the domain ontology, this ontology must be stored too. Registry should also store specifications of composed services (represented as a workflow of other service invocations). Another important functional requirement to the Registry is an existence of management interface to all stored data. This interface should provide a set of methods to add, modify, delete and update Registry data. Particularly, when a new service is registered in the ASG-Registry, Registry stores a new service specification. The platform ontology is also a non-static component. It may evolve over time and thus causes a need to modify and extend ontology data. Figure 2 shows an adaptive process management and the role of Registry in it.

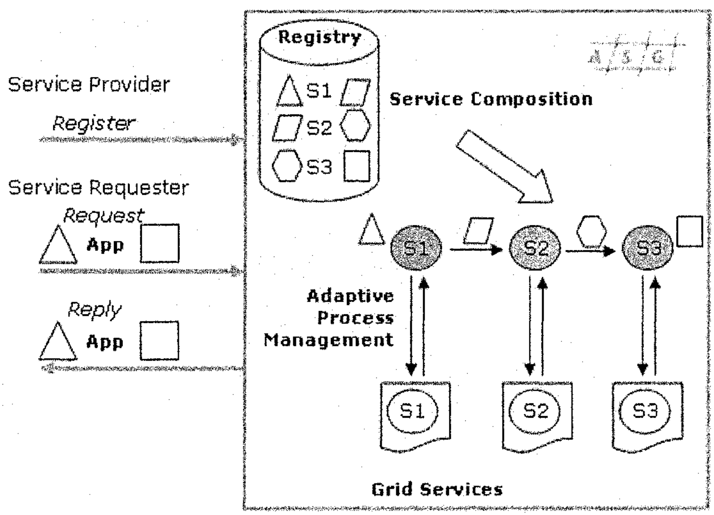


Figure 2. Role of the Registry in composed service invocation in ASG (adopted from [ASG])

From the point of view of non-functional requirements Registry should be a highly reliable and fault-tolerant component. This requirement is crucial, because the ASG-platform architecture implies conceptually centralized and technically distributed solution. Another challenge is scalability. As far as the role of Registry is to store everything about platform data, it becomes necessary to provide a solution capable of managing big data arrays.

ASG-platform requires from Registry to be capable of storing Service Specifications which in turn are done according to ASG-Ontology and refer to domain ontologies of corresponding domains. ASG-Ontology plays a role of a meta-model for Service Specifications. Here further we will refer to a work done by WSMO Working Group [WSMO] towards reusing UDDI as a persistent storage of WSMO-Registry [WSMOReg]. ASG-Ontology is a result of elaboration of WSMO and mappings between them are needed to support specifics for ASG concepts. We consider them both as initiatives facing the same problems of storing (representing) semantically rich data about Web-Services in a Registry-oriented way.

According to the WSMO approach, the data model of UDDI is not extended but necessary WSMO properties are mapped into existing data slots in UDDI (e.g. BusinessEntity) or in customized slots (Identifier bags).

Figure 3 shows the relationships needed to map a WSMO service to a UDDI model. According to proposed approach, existent properties of a UDDI model are reused and WSMO-specific properties are added.

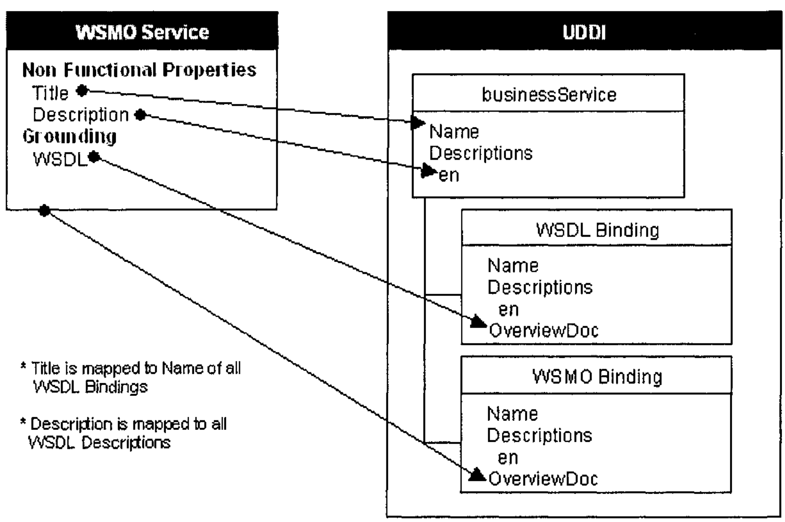


Figure 3. Mapping WSMO Service to UDDI (adopted from [WSMOReg])

4. NEEDS OF SMARTRESOURCE PLATFORM FOR REGISTRY, DISCOVERY AND INTEGRATION

Main source of functional and non-functional requirements for the Global Understanding Environment (GUN) platform is a set of business areas and use cases of the SmartResource project. In addition to existing business use cases from industry, where utilization of the SmartResource platform can be reasonable, the approach of SmartResource introduces new business opportunities and business models of an operation based on an open architecture of the SmartResource platform thanks to open standards of W3C and FIPA, which are the grounds of the platform design.

GUN [Kaykova2005] is a concept used to name a Web-based resource "welfare" environment, which provides a global system for automated "care" over (industrial) Web-resources with the help of heterogeneous, proactive, intelligent and interoperable Web-services. The main players in GUN are the following resources: service consumers (or components of service consumers), service providers (or components of service providers), decision-makers (or components of decision makers). All these resources can be artificial (tangible or intangible) or natural (human or other). It is supposed that the "service consumers" will be able: (a) to proactively monitor own state over time and changing context; (b) to discover appropriate "decision makers" and order from them remote diagnostics of own condition, and then the "decision makers" will automatically decide, which maintenance ("treatment") services are applied to that condition; (c) to discover appropriate "service providers" and order from them the required maintenance. Main layers of the GUN architecture are shown in Figure 4.

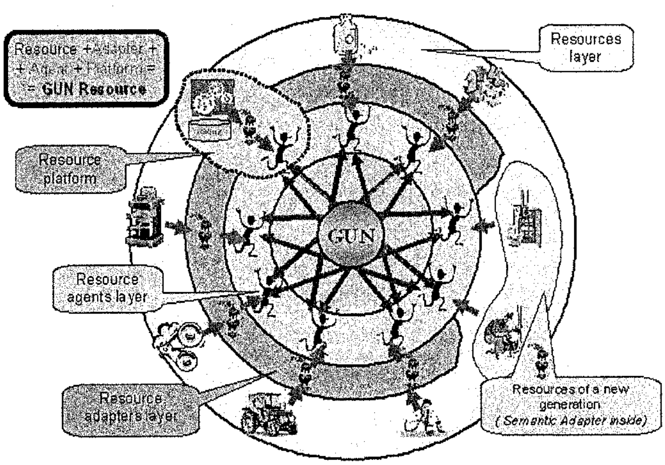


Figure 4. Layers of the GUN architecture

Industrial resources (e.g. devices, experts, software components, etc.) can be linked to the Semantic Web-based environment via adapters (or interfaces), which include (if necessary) sensors with digital output, data structuring (e.g. XML) and semantic adapter components (XML to Semantic Web). Agents are assumed to be assigned to each resource and are able to monitor semantically rich data about states of the resource coming from the adapter, decide if deeper diagnostics of the state is needed, discover other agents in the environment, which represent "decision makers" and exchange information (agent-to-agent communication with semantically enriched content language) to get diagnoses and decide if a maintenance is needed. It is assumed that "decision making" Web-services will be implemented based on various machine learning algorithms and will be able to learn based on samples of data taken from various "service consumers" and labeled by experts. Use of agent technologies within the GUN framework allows mobility of service components between various platforms, decentralized service discovery, utilization of FIPA communication protocols, and MAS-like integration/composition of services.

Condition monitoring of industrial devices is a target domain of the SmartResource project. Research prototype of the GUN environment in this project implements a use case of knowledge transfer from a diagnostic expert to a Web Service with a machine learning algorithm to substitute an expensive human resource by a diagnostic Web Service in the process of condition monitoring. Figure 5 illustrates the use case of a knowledge transfer.

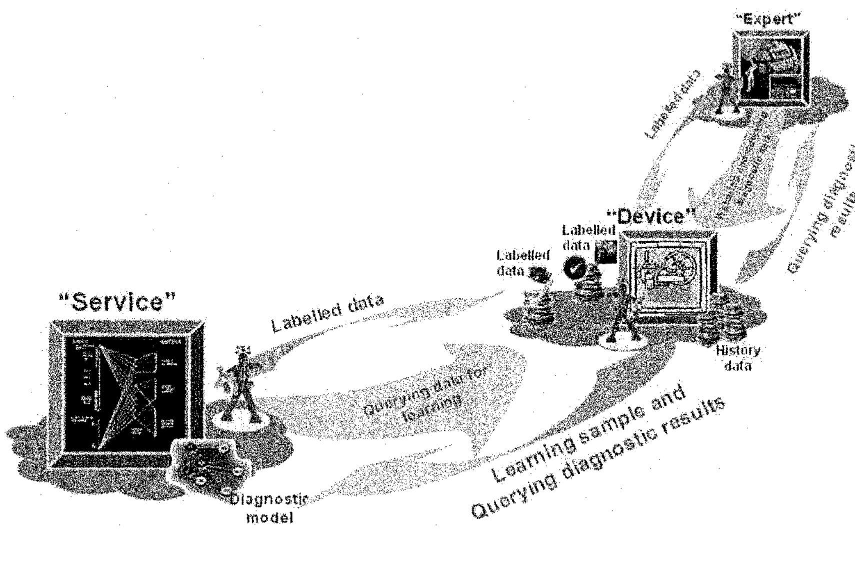


Figure 5. Knowledge transfer use case

However, the condition monitoring use case exists in different domains varying by an object of monitoring. To enable the use cases described above, one of the crucial research and development issues in the SmartResource project is to provide efficient mechanisms of description, discovery and integration of proactive resources.

Semantic Web provides standards for semantic description of resources in the Web that facilitates their discovery and integration. Such kind of a description for resources is called an annotation. RDF, RDFS and OWL cover aspects of a conceptual solution for a meta-data description in a form of ontology and description of a resource as an instance of certain class of resources using facets of this class in a typical case. The SmartResource project treats the entity description as a semantic annotation using Semantic Web standards.

Although an architectural and operational consideration of the process of a resource registration is out of scope of the Semantic Web standards, it is the most important issue for enabling automated discovery and integration. Thanks to the agent-driven platform of GUN, existing tools of multi-agent systems can solve partially tasks of registration, discovery and integration.

Despite of existing tools for multi-agent systems, semantic description, discovery and integration of resources is still an open question. We think that UDDI can be adapted to provide a functionality of a registry or a directory of semantically annotated resources that are SmartResources in the GUN platform.

5. A BRIEF DESCRIPTION OF A UNIVERSAL DISCOVERY OF DESCRIPTIONS AND THEIR INTEGRATION

It is attractive to use existing solutions for registering and discovering instead of implementing something else from scratch. Universal Description Discovery & Integration (UDDI) standard is the first one from the possible candidates, because it is widely spread and supported nowadays by big companies in their commercial and open source Registries.

We think that it is possible to use UDDI 3.0 [UDDISpec] for registration of semantically annotated entities without changes in the specification, while reasoning and other manipulations with the semantics would require changes to the specification of UDDI APIs. Focus of this paper is to provide a solution for mapping from concepts of Semantic Web standards to tModel concepts of UDDI. Possibility for an implementation of an advanced functionality for Registry based on Semantic data without changes to UDDI specification of APIs is also an issue to analyze.

UDDI by definition is a specification of services to provide publishing and discovery of "businesses, organizations and other Web Service providers", their Web Services and technical interfaces to enact those services [UDDIspec].

UDDI specification defines UDDI data model as a format for storing target entities of descriptions. Figure 6 illustrates the UDDI information model. Chapter 6 contains more details about the concept of tModel.

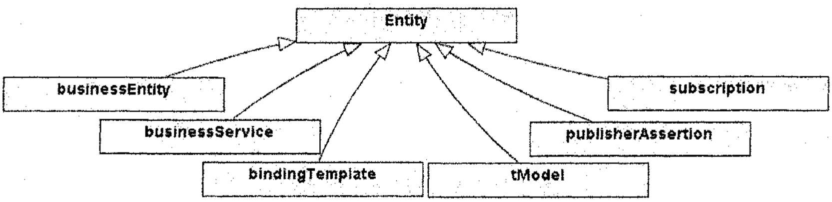


Figure 6. UDDI data model

Figure 7 shows sets of UDDI API defined in the standard.

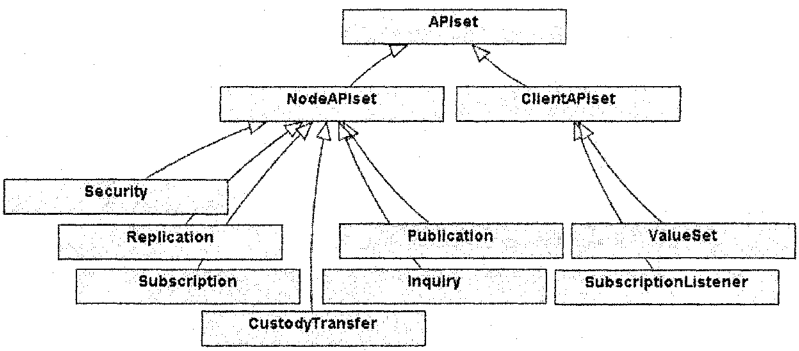


Figure 7. UDDI API sets

Figure 8 summaries the basic architecture of UDDI that allows a UDDI node to be an XML Web Service. The flexibility is achieved because UDDI does not restrict the technologies of the services, about which it stores information or the ways in which that information is decorated with metadata.

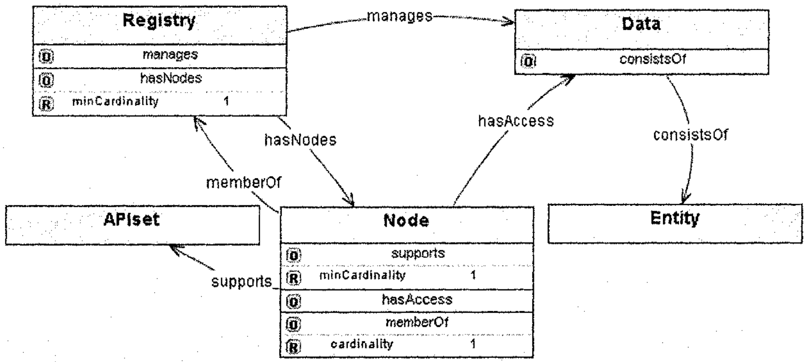


Figure 8. UDDI basic architecture

6. MAPPING OF ONTOLOGY CONCEPTS TO A UDDI DATA MODEL

The main advantage of using UDDI as a basis for an Ontology-based Registry is that a lot of mechanisms needed in Registry (like access rights, an administration, interfaces) are already defined, specified and implemented. Although UDDI does not provide support for semantic search, retrieval and storage, it is already accepted as an industrial standard and a huge number of services already store their service specifications in UDDI.

UDDI model contains a tModel element as a building block for storing different kinds of concepts and relations between them. tModel is a reusable concept, such as a Web service type, a protocol used by Web services, or a category system. The structure of tModel element is presented in Figure 10.

A *TModel* element must contain a name and a description and may contain *tModelKey* as a unique identifier. *tModel* may also have *identifierBag* and *categoryBag* elements. These elements are crucial as building parts of a structure of the ontology storage.

Structure of *identifierBag* is presented in Figure 11.

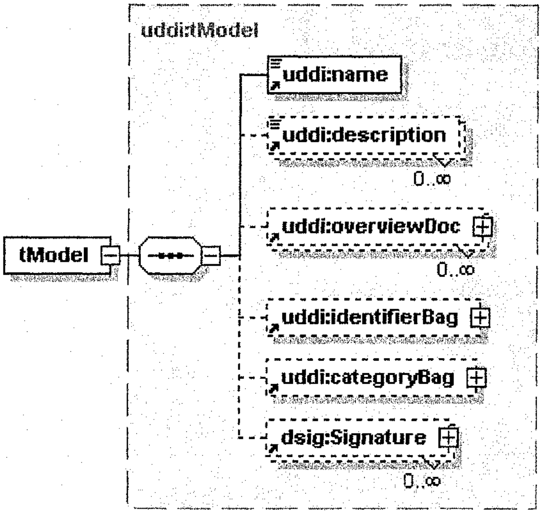


Figure 10. A tModel structure presented using an XML-Schema syntax

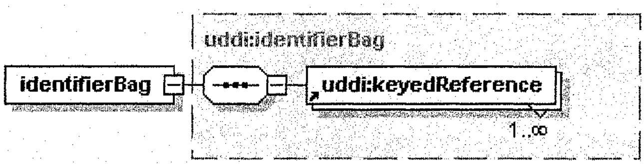


Figure 11. An *identifierBag* element

Thus *identifierBag* may contain 1 or more *keyedReferences*.

The *categoryBag* element has a little bit more sophisticated structure. It allows referring structures to be categorized according to published categorization systems. Figure 12 depicts the *categoryBag's* structure.

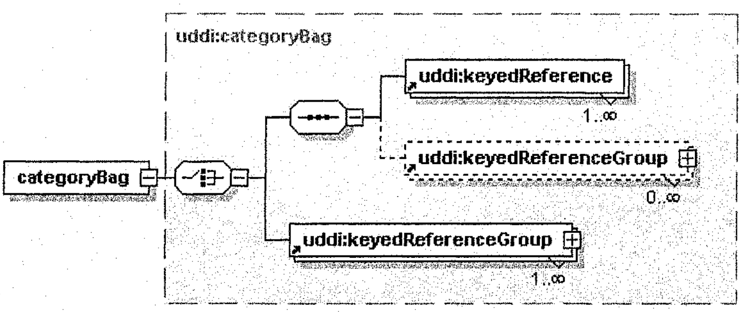


Figure 12. A categoryBag element

categoryBag may also contain directly one or more *keyedReferences*, but also allows a *keyedReferenceGroup* element, which in turn incorporates *keyedReferences* and must contain a *tModelKey* attribute that specifies the structure and meaning of *keyedReferences* contained in the *keyedReferenceGroup* (see Figure 13).

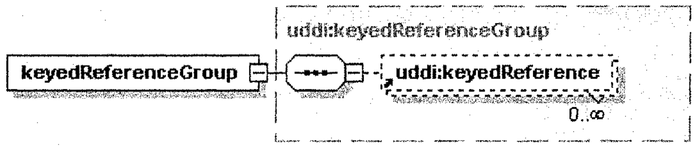


Figure 13. A keyedReference element

A *keyedReference* element, when included in *identifierBag*, represents an identifier of a specific identifier system. The *keyedReference* consists of the three attributes: *tModelKey*, *keyName* and *keyValue*. The required *tModelKey* refers to *tModel* that represents the system of identification, and the required *keyValue* contains the actual identifier within this system. The optional *keyName* may be used to provide a descriptive name for the identifier (see Figure 14).

```
<identifierBag>
    <keyedReference
        tModelKey="uddi:someidentifier"
        keyName="some descriptive name"
        keyValue="some value" />
</identifierBag>
```

Figure 14. Example of *keyedReference*

How this structure can be used to represent an ontology? Below we will consider a couple of examples of its application.

Example 1. *Ontology description language as a generic concept.*

In this approach we introduce a generic concept, e.g. "RDF Schema" and refer to it as to a set of concepts contained in RDF-Schema (Figure 15).

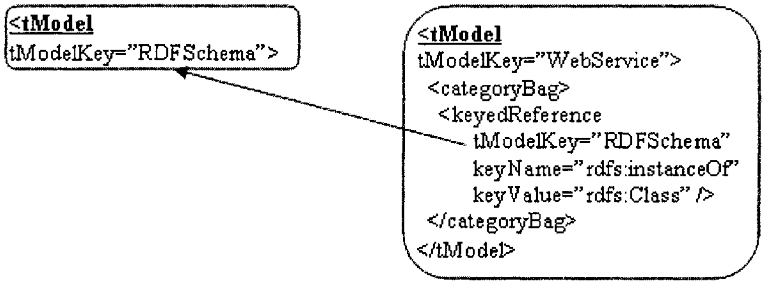


Figure 15. Introducing RDF-Schema as a tModel

After a quick view it looks reasonable to reuse a *keyName* element for defining the relationship between class *WebService* and the RDF-Schema concept *rdfs:Class*, referred by the *keyValue* element. However, when we want to create a subclass of a WebService class, say, *SemanticWebService*, then how to decide to which categorization scheme to refer? If we define tModel in a similar way (see Figure 16), we make a conceptual mistake, because *keyValue* of *keyedReference* has to point to a concept belonging to the set of concepts, defined by the referred *tModel*. In other words, *SemanticWebService* does not belong to the element set of "RDFSchema".

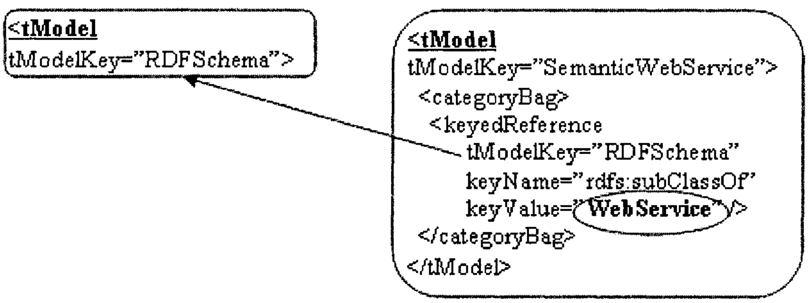


Figure 16. Wrong reference to the RDFSchema tModel

In this example we comply with the UDDI syntax, but contradict the semantics of UDDI concepts. It means that data stored according to this way will not be reusable by standard UDDI searching facilities.

Example 2. *Introducing concepts explicitly.*

In this case we reuse UDDI structures to represent classical relationship "*subject-predicate-object*" and build the ontology model on top of it. We map "*subject-predicate-object*" construction to the UDDI construction "*tModelKey-tModelKey-keyValue*" (see Figure 17).

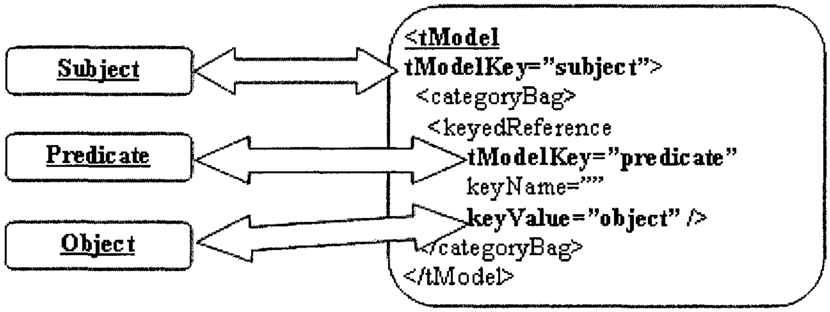


Figure 17. Concepts mapping

Let us continue with an RDF-Schema example. First of all we need to define RDF-Schema concepts as UDDI *tModels* (see Figure 18).

```
<tModel
tModelKey="rdfs:Class">
  <categoryBag>
    <keyedReference
      tModelKey="rdfs:subClassOf"
      keyName="subClassOf relation"
      keyValue="rdfs:Resource" />
    <keyedReference
      tModelKey="rdf:type"
      keyName="type relation"
      keyValue="rdfs:Class" />
  </categoryBag>
</tModel>
```

```
<tModel
tModelKey="rdf:Property">
  <categoryBag>
    <keyedReference
      tModelKey="rdfs: subClassOf"
      keyName="subClassOf relation"
      keyValue="rdfs:Resource" />
    <keyedReference
      tModelKey="rdf:type"
      keyName="type relation"
      keyValue="rdfs:Class" />
  </categoryBag>
</tModel>
```

```
<tModel
tModelKey="rdfs:Resource">
  <categoryBag>
    <keyedReference
      tModelKey="rdf:type"
      keyName="type relation"
      keyValue="rdfs:Class" />
  </categoryBag>
</tModel>
```

```
<tModel
tModelKey=" rdf:type">
  <categoryBag>
    <keyedReference
      tModelKey="rdf:type"
      keyName="type relation"
      keyValue="rdf:Property" />
  </categoryBag>
</tModel>
```

```
<tModel
tModelKey=" rdfs:subClassOf">
  <categoryBag>
    <keyedReference
      tModelKey="rdf:type"
      keyName="type relation"
      keyValue="rdf:Property" />
  </categoryBag>
</tModel>
```

Figure 18. Definition of RDF-Schema concepts

As far as RDF-Schema is quite a big document, all concept mappings will not fit in the size of this paper, so we define only the crucial elements for our example. Based on the above upper-level structure now we can try to model the same concepts from Example 1, but referring to a new structure (see Figure 19). This description does not contradict UDDI semantics, because *keyedRefernce's tModelKey* referring to a property assumes that all *keyValues* of *keyedRefernces* with the same *tModelKey* belong to the set of objects referred by this property.

```
<tModel
tModelKey="WebService">
  <categoryBag>
    <keyedReference
        tModelKey="rdfs:subClassOf"
        keyName="subClassOf relation"
        keyValue="rdfs:Resource" />
    <keyedReference
        tModelKey="rdf:type"
        keyName="type relation"
        keyValue="rdfs:Class" />
  </categoryBag>
</tModel>
```

```
<tModel
tModelKey="SemanticWebService">
  <categoryBag>
    <keyedReference
        tModelKey="rdfs: subClassOf"
        keyName="subClassOf relation"
        keyValue="WebService" />
    <keyedReference
        tModelKey="rdf:type"
        keyName="type relation"
        keyValue="rdfs:Class" />
  </categoryBag>
</tModel>
```

Figure 19. Definition of classes

7. CONCLUSIONS

The main goal of this paper was the evaluation of UDDI capabilities to store semantic descriptions of entities to enable semantic discovery and their integration with Semantic Registries in future based on mature and widely adopted UDDI specifications. Our conclusion is that UDDI as such provides enough support for registration of semantically annotated resources, but additional efforts are needed to elaborate API to support a semantic discovery of registered resources.

We have presented an approach of mapping RDFS upper concepts to a UDDI data model using a tModel structure.

While other publications in the area of enhancement of UDDI with semantics consider mainly semantic discovery and relevant enabling architectures, we have an opinion that the challenge of publishing semantic annotations of resources in UDDI has to be met without changing the UDDI architecture and APIs to enable semantic queries.

We consider two on-going projects (ASG and SmartResource) as the use cases and we have shown that publishing an ASG service and a domain ontology into UDDI can be performed based on mapping WSMO to a UDDI information model. The SmartResource project could use UDDI to implement Notice Boards for registering semantically annotated resources in a P2P environment. In parallel to a practical utilization of UDDI in these two research projects, further research is needed to elaborate semantic discovery algorithms and APIs of UDDI based on the proposed way of storing semantics in UDDI.

8. ACKNOWLEDGEMENT

This research has been supported partly by the "Proactive Self-Maintained Resources in Semantic Web" (SmartResource) project funded by TEKES and the industrial consortium of Metso Automation, TeliaSonera, TietoEnator and Science Park and partly by the "Adaptive Services Grid" (ASG) 6th Framework Integrated Project (EU-IST-004617).

9. REFERENCES

[ASG] "Adaptive Services Grid", Integrated project supported by European Commission, http://asg-platform.org/

[DAML] "DAML Language", http://www.daml.org/language/

[DAML-S] "DAML Services", http://www.daml.org/services/

[FIPAAA] "FIPA Abstract Architecture Specification", http://fipa.org/specs/fipa00001/

[FIPAAM] "FIPA Agent Management Specification", http://fipa.org/specs/fipa00023/

[Kaikova2004] Kaikova H., Khriyenko O., Kononenko O., Terziyan V., Zharko A., Proactive Self-Maintained Resources in Semantic Web, Eastern-European Journal of Enterprise Technologies, Vol. 2, No. 1, 2004, pp. 4-16.

[Kaykova2005] Kaykova O., Khriyenko O., Kovtun D., Naumenko A., Terziyan V., Zharko A., General Adaptation Framework: Enabling Interoperability for Industrial Web Resources, In: International Journal on Semantic Web and Information Systems, Vol. 1, No 3, 2005, Idea Group, pp. 30-62.

[Kawamura2004] T. Kawamura, J. D. Blasio, T. Hasegawa, M. Paolucci, K. Sycara, "Public Deployment of Semantic Service Matchmaker with UDDI Business Registry", Proceedings of 3rd International Semantic Web Conference (ISWC 2004), LNCS 3298, pp. 752-766, 2004.

[Kawamura2003] T. Kawamura, J. D. Blasio, T. Hasegawa, M. Paolucci, K. Sycara, "Preliminary Report of Public Experiment of Semantic Service Matchmaker with UDDI Business Registry", Proceedings of First

International Conference on Service Oriented Computing (ICSOC 2003), LNCS No. 2910, pp. 208-224, 2003.

[Moreau2005] L. Moreau, S. Miles, J. Papay, K. Decker, T. Payne, "Publishing Semantic Descriptions of Services", Semantic Grid Workshop at GGF9, 2005.

[OWL] "Web Ontology Language", http://www.w3.org/2004/OWL/.

[OWLS] Ontology Web Language for Web Services (OWL-S) 1.0, http://www.daml.org/services/owl-s/1.0/.

[Paolucci1] M. Paolucci, T. Kawamura, T.R. Payne, K. Sycara, "Semantic Matching of Web Services capabilities", Proceedings of First International Semantic Web Conference (ISWC 2002), IEEE, pp. 333-347, 2002

[Paolucci2] M. Paolucci, T. Kawamura, T.R. Payne, K. Sycara, "Importing the Semantic Web in UDDI", Proceedings of E-Services Semantic Web Workshop (ESSW 2002), 2002.

[Pokraev2003] S. Pokraev, J. Koolwaaij, M. Wibbels, "Extending UDDI with Context-Aware Features Based on Semantic Service Descriptions", Proceedings of the International Conference on Web Services, ICWS '03, June 23 - 26, 2003, Las Vegas, Nevada, USA. CSREA Press 2003, ISBN 1-892512-49-1, pp. 184-190.

[Powers2003] Shelley Powers, Practical RDF, O'Reilly, July 2003, ISBN: 0-596-00263-7, 350 pages.

[RDF] "Resource Description Framework", http://www.w3.org/RDF/.

[RDFS] "RDF Vocabulary Description Language 1.0: RDF Schema", http://www.w3.org/TR/rdf-schema/.

[RuleML] "The Rule Markup Initiative", http://www.dfki.uni-kl.de/ruleml/.

[SemanticWeb] "Semantic Web", http://www.w3.org/2001/sw/.

[SmartResource] "SmartResource: Proactive Self-Maintained Resources in Semantic Web" project, http://www.cs.jyu.fi/ai/OntoGroup/SmartResource_details.htm

[Srinivasan2004] N. Srinivasan, M. Paolucci, K. Sycara, "An Efficient Algorithm for OWL-S Based Semantic Search in UDDI" Semantic Web Services and Web Process Composition, First International Workshop, SWSWPC 2004: 96-110.

[UDDIspec] UDDI Specifications, http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm.

[UDDI] "Universal Description, Discovery and Integration", http://www.uddi.org/.

[WSMO] "Web Services Modeling Ontology", http://www.wsmo.org.

[WSMOReg] "WSMO Registry", Working Draft, http://www.wsmo.org/2004/d10/v0.1/.

[WSDL] "Web Service Definition Language", http://www.w3.org/TR/wsdl.

[XML] Extensible Markup Language specification site, http://www.w3c.org/XML/.

ON THE ROAD TO BUSINESS APPLICATIONS OF SEMANTIC WEB TECHNOLOGY

Sematic Web in Business - How to Proceed

Kari Oinonen

Kiertotie 14 as 3, 40250 Jyväskylä, e-mail: kariroinonen@kolumbus.fi

Abstract: This paper discusses potential usage of Semantic Web in business applications and provides one way to proceed faster.

Current situation in Semantic Web application area is discussed. General appropriate trends in technology development and in communication industry and industry general are reviewed in order to see how Semantic Web technology fits with these trends.

Finally this paper suggests that, to get the technology into use, a common application framework should be formed. This framework shall look not only the technology, but also application, ICT architectures and business models this technology makes relevant. Definition of this framework is proposed to be a part of a road map process for which guidelines are provided.

Key words: road map, Semantic Web , semantic technology, business models, ICT architecture

1. BUSINESS APPLICATIONS AND SEMANTIC WEB

The potential of semantic technology is far too wide to be covered fully. There exists Semantic Web applications like MuseumFinland in Semantic Web (MuseumFinland) or some e.g. diagnostic applications in production or preproduction stage. However it is a fact that Semantic Web applications have emerged much slower pace than was assumed by field or technology experts. Not in great numbers can we see application types and application areas where the semantic layer in the Web can make formerly impossible things possible, formerly uneconomical things and applications economical. This may be explained partly by the fact that some successful applications are not publicly known. Meanwhile various kind of more traditional Web applications are being created in big numbers. Why is that. The next section discusses on this more deeply.

1.1 Why Semantic Web has progressed so slowly

The following list describes some of the facts that have or are affecting on the progress of Semantic Web and its applications in business. One of the main facts is that the problems related to Semantic Web are one or two orders in magnitude bigger compared to problems that were encountered with the Web. This is because of the complexity involved with the semantics and communication in general.

- Technology development has taken time and resulted in changes in tools and standards.
- Standardization: problems with standardization and parallel development, perhaps too many standards so users do not know which ones to use for what.
- Competition within technology.
- Distribution of development.
- Possibly the existing Semantic Web technology do not cover the problem or business needs area well enough.
- Real existing business needs are lacking or are not recognized.
- Too complex (technology).
- Too new – not proved.
- Most public examples are more or less simple ones and they may not reflect the technology potential.
- Existing applications and business models are naturally something newer technology and business models have to overcome.
- Semantic Web technology may be at best in areas that have not yet been tried enough. I mean communication between applications or humans who have not been able to communicate because too high costs or too complex integration and associated maintenance costs involved with the ICT systems.
- Currently - and probably in the future - user base only fraction of HTML user base.
- People who understand the HTML world in general are countless. Compared to the basic Web technology gaining corresponding competence and understanding level of semantic web technology and

related terms and concepts is a much bigger task. Only a fraction of people can navigate with these concepts or use corresponding tools.

- Another challenge for application of the technology is the lacking of application domain expertise that is needed to bring together with the semantic technology. From the technology point of view this may be the biggest cause of delay that is still affecting the technology progress to business applications.

Just looking the web pages of W3C (W3C) reveals part of the problem. Semantic Web is just one topic among more than 50 topics. It is not a straightforward task for a newcomer or potential future user or decision maker to find out relations between these topics and their applicability. The another part of the problem is not a technical one. A national research project RUBIC (RUBIC) pointed out that technology is not the main limiting factor in interoperability and networking between companies. The main challenge is how to get companies into open and co-operative development of business models, business processes and supporting technologies.

This paper aims to overcome some of these causes of delay or obstacles by suggesting a co-operation between relevant parties to form a road map on how best utilize possibilities the Semantic Web enables. The suggested road map will also serve as a means to rise the level of understanding the technology potential for decision makers in the businesses. The end part of this paper provides a process according to which a road map can be formed.

2. LOOKING HOW GENERAL TRENDS RELATE TO WEB AND ESPECIALLY SEMANTIC WEB POTENTIALLY INCREASING USE

Here I will cover some of the facts that have affected in the adoption of the Web in its basic form and trends that are affecting or can affect on the adoption of the semantic layer on the Web.

2.1 General technology trends

According to Altchuller there are eight factors along which technology develops. These are (Altchuller, G. 1998): life-cycle, dynamization, multiplication cycle (transition from mono- to bi- and poly- systems, transition from macro to micro level, synchronization, scaling up or down, uneven development of parts, replacement of human (automation).

Looking the list above we can note that the basic Web technology has taken the development further by increased possibilities to manage life-cycles in business, there has been an increase in dynamization of some of the content on the Web, linking mechanisms have provided possibilities to combine file based information with information in other files. The content of Web pages have become more and more structured as XML-tagging is being taken into use widely. And what is important to note is that the amount of human work needed is decreasing in proportion to automation in areas like e-business and automated content configuration or filtering.

When comparing Semantic Web potential to the current Web technology, it is possible to note the following: semantic layer is promised to offer functionalities that increase dynamization, takes multiplication cycle further, enables management of increasingly more detailed information pieces, enables synchronization of content and events as time based content management becomes a possibility, and takes the automation level higher. In that respect the Semantic Web can be seen to lie in a natural development path of the Web.

2.2 Changes in communication technology

Looking to the past not log ago we can note that communication and Web based information systems are more and more common. Today it may be difficult to find a newer ICT system, which does not have a some kind of Web connection.

In the telecommunication business the technologies are converging. At general level this means that more and more of the voice and data contents are being delivered via Internet infrastructure. One of the key parameters in this convergence is the fact that Internet infrastructure is relatively cheap. Transmission speeds of Internet are increasing faster than in other technologies in telecommunication. This fact is a strong favor for more and more content being transferred between devices via Internet. Even mobile devices are starting to support Web.

The convergence taking place in communication industry opens up many doors and application areas to be utilized by semantic technologies. This means that Web technologies and Semantic Web technologies can be applied more and more easily on the content that is currently mainly managed in telecommunication networks.

General increase of information structuring – textual and audiovisual – is increasing. Not perhaps the proportion of structured compared to non-structured but still more and more content is being tagged in HTML and increasingly in XML and this trend is continuing at least where content re-use and configurability are of importance.

2.3 Changes in business needs

One special feature that has changed recently in many business areas is the fact that there is an increasing need to be able to look complex topics from different points of view. Depending on the industry there are needs of information management supporting engineering which itself consists of many disciplines, asset management, maintenance management, quality reliability and environmental information management. Traditionally the solution has been several independent IT system solutions. However that kind of solutions have resulted in risks of local optimization and decrease on competitiveness. From the business point of view there exists heavy interdependencies and needs of data access between business functions or sub-businesses.

Related to this, knowledge and services based businesses, which are gaining favor, rely on capabilities to manage information which is essentially of network type and is originally created in a heterogeneous ICT environment.

3. ON THE NEED OF GENERAL APPLICATION AND BUSINESS FRAMEWORK

Benefits of creating a general application and usage oriented road map for the Semantic Web lie in two areas. One area where a road map is useful is interoperability of technologies and applications. Without general framework especially applications will not be inter operable. This means that that semantic layer of the Web will not be a semantic web. Instead, there will be a collection of application specific solutions and technologies which are poorly inter operable. That kind of situation will not support semantic web to the main road of technology. The other area where the road map is essential is to provide possible future users and decision makers means to provide general understanding of the technology, what the technology can do for them and when it can do that. These are the main reasons the general road map process is proposed.

The Web in the form most of people understand in the WWW started from the need. The need was to overcome the obstacle of being able to read fellow workers publications and project reports in CERN. To overcome this a simple page layout description language HTML was developed. Most of the additional technology needed was in place already. HTML proved functional and turned to be a success – a bigger success than anybody could imagine. Now we must again recall what was the original need. It was the need to be able to read fellow worker's publications and reports that were originally produced by one of the several writing tools in use at CERN. Layout oriented HTML was the common layout language for documents, HTML viewers provided viewing capability, file access and transfer were made possible by http based on ip-protocol. There was no need for road maps or general application framework as the original problem was solved.

How the previous paragraph relates to the Semantic Web development. There is an analogy: Semantic Web seems to be developed to add WWW capabilities to actively search and retrieve and access data on the Web by active agents by utilizing semantic meta data. This in turn would increase resource use and interoperability of the data in the Web. Looking this initiative and comparing it with the discovery of WWW reveals that: The problem is analogous, but far more general. The problem itself is or seems to be much more complex than the problem of read-accessing fellow workers file contents. The problem itself is not so clear – it seems – even to the developers and this leads to a situation where the Semantic Web is being developed more on the technology aspect and the real world business needs are somewhat in the background.

The development of the Semantic Web and related tools are technology and solution driven. This is not not necessarily a handicap, if like was the case with HTML, the technology is targeted to solve the real and right problems. The problem for business applicability rises in this case from the fact that there exists some basic semantic technology, some of the technology needed may even be missing, the time span of the development is wide and still there seems to be no common understanding how to use the technology. There exists fully functioning semantic and Semantic Web applications, but often these applications are specially built for a purpose leading to a situation where these Semantic Web applications are in essence not inter operative.

Much of the technology and application development is taking place in distributed and even competitive environment. Continuing this route leads to non-convergent technology and slowing down of the technology adaptation and use. The fact that various semantic web technologies and applications are not inter operative is one of the key issues this paper aims to overcome. If development resources are needed to be utilized effectively and if the technology potential is needed to take into use, then a plan or road map is useful. The road map can act as a rough plan or guideline for the semantic technology adaptation and development too. In essence a road map makes the field more visible to the technology developers, financial decision makers and future users. Road map can provide also guidelines on what technology or standards may best be applied and where and describe some future application areas that do not currently exist.

What makes the need for the semantic technology and especially Semantic Web application development and usage even more challenging is the combination of the following three facts:

- The definition of ontology: ontology is a formalization of conceptualization (Heimbürger, A. et al. 2004). Formalization itself is a straightforward task. The conceptualization process is instead a very challenging one. Challenges in the conceptualization process get bigger as the potential ontology user group gets bigger. The conceptualization needs agreement of potential users either in the ontology development phase or later phase when the ontology is taken into use. In both phases mapping between existing more or less local or proprietary concepts and concepts in the ontology needs to be done.
- Ontology usage in the Semantic Web: ontologies are treaded as global. This is fine if we can agree on the ontologies on some global forum. In general this is not possible although there exists limited area globally applicable ontologies. The need to be able at first stage to agree on ontologies is a challenge to overcome in many real world business cases. Of course the communication and automated resource discovery is eased if there is an agreement on terms concepts and their relations agreed beforehand. In general communication case this is not possible.
- The intention of the Semantic Web is: "The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in co-operation" (Berners-Lee, T. et al. 2001), then the way ontologies are created and used in Semantic Web only provide capability to very limited resource discovery and access.

If the limitation imposed by centralized architecture of ontologies management is of importance or not depends entirely on the business case – how the Semantic Web needs or is wanted to be used. In a machine diagnostic case it is well possible to create suitable ontologies mainly for own purposes and arrange limited but functioning interoperability between devices to be monitored on the field and a diagnostic service provider. For that particular service provider it is enough to use their own ontology. On the other hand the same equipment or device or plant specific information is used in documentation creation and maintenance, at design and maintenance stages in life-cycle. The problem of interoperability and information integrability is partly transferred to the ontology level.

Trust in the technology is essential for wide and general business adoption. Trust is related to the technology itself, its applicability, costs and benefits that are gained. The other part of trust is related to information security. In the Web much of the data is available to be browsed. Password protection and similar means are used to limit the accessibility of the Web pages to only those that are allowed. In Semantic Web the data, information and knowledge is in a more structured and thus generally more detailed form compared to a Web page of file level. With the exception that in semantic environment more meta data may be available. To take the decision of how inter operative we want the semantic layer of the Web to be, we have to consider what data, information or knowledge is wanted to be inter operative. Are there needs in favor of general inter operativeness, or is it enough that Semantic Web will be used mainly for point-to-point information integrations purposes. If interoperability is wanted to cover, say a certain pool of data, then the technology should allow this even if this pool of data was formed grouping together data described by several independent ontologies.

In the Semantic Web environment new kinds of security mechanisms are needed to guarantee access to those allowed and at the same time protect valuable business information and knowledge. One of the solutions may come from the separation of security issues from the actual information content or service provision server.

Data and information encryption is one key topic that a road map should look at. This is because instead of accessing data for reading we are entering to a more dynamic situation where data integrity must be able to guarantee. Password protection will not be flexible enough on situations where interoperability takes place over several independent protected and unprotected systems.

A road map plan is provided for the Semantic Web. Road map process is presented with topics. The motivation for making the road map is to speed the pace the semantic technology is taken to the business use. Besides the main topics and propagation order suggestions there are suggestions who should contribute to the road map. In essence the road map definition process itself is of great importance because the process produces not only the road map but also creates understanding and wider view to the technology and its business use among the parties involved in the process

4. ROAD MAP PLAN

The Semantic Web road map plan consists of two main sections or parts. The first part defines the road map definition process. The second part helps in reaching agreement on the Semantic Web vision in terms of business, business models, information and knowledge management requirements and technology usage in the area. The road map process provides the Semantic Web vision definition and will be a definition of the framework for Semantic Web business usage.

4.1 Road map definition process

There are several sources to get an applicable road map process and associated other information needed. The following description is based loosely on (Naumanen, M. 2001) and a process used in national TEKES technology development program (ÄLY 2004). It can be stated that road maps have been formed in Finland lately in many technology areas and there are some clear benefits of defining a road map for a field. One of the main benefits is a more unified view of the field that is created among stakeholders. To achieve this the Semantic Web road map should be published in appropriate forums.

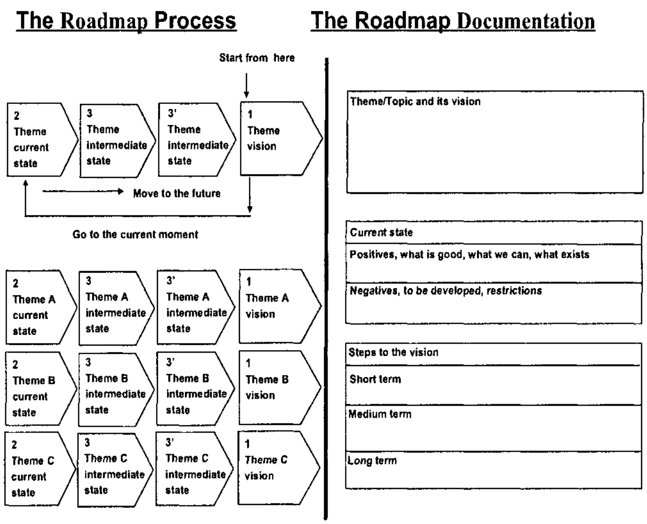


Figure 1. The Road map process and documentation

Generally the topics for the road map can be selected freely but often topics or views of market and business, applications and services, technology and R&D projects and science are used. The major task is to form development paths of the topics and to define what are the interactions of these topics.

What will make the road map process actualization a challenging task in the case of Semantic Web , is the fact that the area to be covered is huge. So, in that respect there exists no single body that can form the road map and run the whole road map process taking the following list into account: the road map should cover many different business areas, it should cover many business models from single part producers to service business, the technology aspects and related existing standards and models usage and finally the one of the essential thing for communication: common language - ontologies management. Therefore it is suggested that there will be a suitable division of small dynamic work groups that contribute to the road map. Business and industry should take part in the work for two essential reasons: correct requirements and buy-in.

4.2 Road map process

In the following a collection of some essential Semantic Web road map topics are presented. For each topic there are presented some subtopics that I feel has to be covered in the semantic web road map process. Together the topics and their subtopics form the basic requirements that a road map for Semantic Web in business use shall cover.

As the road map process presented earlier describes, it is suggested that the vision is defined first. The Semantic Web vision in the W3C (Berners-Lee, T. et al. 2001) can be used as the basis. To support the Semantic Web vision statement it is beneficial to form a description of what can be done with the technology when the vision is true. Additionally it is needed to have technology, use cases, business and user needs and ontology management defined at vision level. These vision definitions can then guide the path definition during the road map work.

As an example we could assume that the vision is true after 10 years of time in a board context. Starting from the vision and its description we come to the current state. Excellent review depicting current state and needs in business communication and possibilities for semantic technology applications was formed by national project RUBIC (RUBIC). This project concentrated on interfaces between companies or between applications. Not all possible application areas for the semantic web were covered. However RUBIC project results can be used as a good starting point. As the first intermediate state we can choose two years and for the next intermediate we can choose five years from now. For these intermediate steps that lead to the vision we need to define and describe what is possible with the technology and how it is used in each particular state.

4.2.1 Technology topic

This topic in the road map covers technology issues. What Internet and Semantic Web technologies are ready and when, what technologies may be best applied on what application areas and how to guarantee interoperability.

- What Semantic Web standards to use on what purpose : SUO (standard upper ontology), DAML (www.daml.org), OIL (http://www.ontoknowledge.org/oil/), OWL, PSL , KIF, etc.
- Relations and connect ability to e.g. STEP (ISO 10303) standards, national or international industry specific standardization initiatives on procurement item management and on electrification projecting (PSK 7401) data exchange.
- What kind of information management functionalities and capabilities are needed in e.g. adaptability of business networking, life-cycle support of products and services – from the requirements management to the usage and service phases, innovation process, competence management, e-business and relations to standards in e-business management – ebXML and RosettaNet.
- Semantic Web relation and support of modeling activities like CIM-OSA, or MFM.
- How to manage the fact that industry areas are at different development stages.
- There should be a review to check usability of existing ontologies and the need to create ontologies from needs basis.
- How to guarantee future compatibility if the technology and ontologies are developed concurrently and parallel.
- User interfaces for the semantic multidimensional information.
- How to guarantee information integrity and security in a distributed heterogeneous environment.
- What other technology can be beneficial to join together with semantic technology or semantic information for a certain kind of task or application. As an example Web Services provide handy means for semantically marked information access.

4.2.2 Use cases -topic

A list of basic functionality Semantic Web technology is presented. Some of the listed functions are basic functions of the technology and some are built on the basic functions. The question is of communication between IT systems or applications, but with the aid of suitable devices this can be extended to man-to-man communication and to other combinations.

Initiatives like the W3C Speech Interface with a suite of markup definitions are aiming to realize this kind of communication combinations (W3C Speech Interface).

- Finding of information and resources.
- Building of collections of interdependent Web pages.
- Classification of information.
- Information collection for analysis or decision purposes.
- Linking of information in Web pages between different information locations.
- Maintenance of the links between pieces of information.
- Support of network type of information models.
- Information analysis based on other information or knowledge.
- Knowledge management based on the basic functionality.

4.2.3 Business and usage -topic

This topic shall act as needs and requirements for other topics. General and industry specific trends should be reviewed as well as industry and business representatives shall be interviewed to get better insight of business requirements and the change of requirements during 10 years time span. The road map should clearly point out how business needs relate to Semantic Web technology and its matureness at current state, at each intermediate state and at the target state 10 years from now.

- The meaning of Semantic Web usage to the customer or end user in terms of costs, speed and benefits in information and knowledge management.
- The effects of Semantic Web on knowledge and service based businesses.
- To be taken into consideration: different industry and business areas have their own needs and preferences that can have affect on the development and the pace Semantic Web is taken into use within that industry or business.
- What kind of functionality and capabilities are needed in e.g. adaptability of business networking.
- Life-cycle support of products and services – from the requirements management to the usage and service phases, innovation process, competence management, e-business and relations to standards in e-business management – ebXML and RosettaNet (RosettaNet).
- In life-cycle support and in related knowledge and service business it is a need to be able to support and manage network type of information models.

- Current and future needs of industry or business that are potential users. Find out at general level needs change in 3, 5 and 10 years perspective to help targeting.
- The challenges of managing effectively ever increasing number of documented and non-documented information is increasing continuously.
- Definition of the most promising application areas from the business point of view.

4.2.4 Ontology topic

Ontologies and their management is one of the key components of the Semantic Web. How ontologies are created, managed and maintained and how easy it is to integrate and refer other ontologies defines ontological limitations to the success and interoperability of the whole semantic web.

- There should be a review to check usability of existing and applicable ontologies and the need to create ontologies from needs basis.
- Different industry and business areas have their own terms and concepts in use. There are limitations on how well it is possible to force a certain ontology into use.
- Ontologies management: the top-down approach is not a feasible except in limited areas. There are industry or industry area specific initiatives that have produced applicable basis for the formation of ontologies. These can be utilized. There are existing generally accepted standards and ontologies that are widely used. Example of these is Dublin-Gore (Dublin-Core) in publication business and RosettaNet in e-business (RosettaNet).
- Currently ontologies are treated as global, but in practice there is a need for smaller, more local, more dynamic and at the same time inter operable ontologies.
- If it is concluded, during the Semantic Web road map process that a more dynamic way of ontologies management is needed, then there must be mechanisms to update ontologies and state the ontology owner. As these are very important meta data concerning the ontologies them self.
- Complexity can be managed by organizing the elements into local networks or modules which, because of their connectivity, have strong, well defined behavioral characteristics (Tossavainen, T. 2002). This reduces the global burden of producing coherent behavior, since the internal behavioral co-ordination of the modules is substantially handled locally. For ontologies this principle means distributed nature of ontologies development and management. Also this means that there should be mechanisms that guarantee good enough interoperability between ontologies where a need exists.

4.2.5 Who should contribute

As the road map definition process for the Semantic Web and its business use is a challenge because of the nature of the task, it is not possible that a small group of people can define the road map. Instead there is a need for co-operation between technology developers in research institutes, universities and companies; ICT managers and business developers as well end users in business; experts in standardization bodies and technology integrators.

5. HOW TO CONTINUE

One possibility to proceed is to form a small group of experts who could at first stage predefine a suggestion of the vision together with industry and research representatives. After that a more formal and wider road map definition process could be started. The road map process should be finalized and results published in less than a year.

6. REFERENCES

Altchuller, G. 1998. *40 Principles; TRIZ keys to technical innovation.* Technical innovation center, Worchester, MA, 1997.

Berners-Lee, T., Hender, J., Lassila, O., The Semantic Web, Scientific American, May 2001.

Dublin-Core, Dublin-Core Metadata Initiative, http://dublincore.org, (referred 29.4.2005).

Heimbürger,A et al.(2004) . *XLinkTime: Developing time-sensitive linking mechanisms for the future Web.* Julkaisusarja A – Tampereen teknillinen yliopisto, Porin yksikkö. Nro A4/2004.

ISO 10303, http://www.tc184-sc4.org/SC4_open/SC4_Work_Products_Documents/STEP_(10303)/.

MuseumFinland – Finnish Museums on the Semantic Web, March 8, 2004. University of Helsinki, Department of Computer Science, Helsinki Institute of Information Technology, http://museosuomi.cs.helsinki.fi (referred 29.4.2005).

Naumanen, M., *Roadmap – Kartta Menestykseen*, published by Teknologiainfo Teknova Oy, 2001, ISBN 951-817-775-9.

PSK 7401, Information Exchange in Electrification and Automation Project, http://www.psk-standardisointi.fi/Standardiluettelo/tekstiSS.htm. (referred 29.4.2005).

RosettaNet, http://www.rosettanet.org (referred 13.6.2005)

RUBIC, http://www.tieke.fi/mp/db/file_library/x/IMG/13108/file/Rubic_Loppuraportti_V3.0_02_07_04.pdf , in Finnish, (referred 13.6.2005)

Tossavainen, T., *System Usability of Complex Technical Systems,* Acta Polytechnica Scandinavia, Industrial Management and Business Administration Series No. 12, Espoo 2002, 236 +17 pp. Published by the Finnish Academies of Technology. ISBN 951-666-597-7, ISSN 1459-9426.

ÄLY, TEKES technology program, http://webserv2.tekes.fi/opencms/opencms/OhjelmaPortaali/Kaynnissa/ALY/fi/julkaisutjalinkit.html. (referred 29.4.2005).

W3C, http://www.w3.org/ (referred 13.6.2005)

W3C Speech Interface, http://www.w3.org/Voice/Activity.html (referred 13.6.2005)

RFID-based Logistics Information Service with Semantic Web

Dae-Won Park[1] and Hyuk-Chul Kwon[2]

[1] Department of Computer Science, Pusan National University,
Busan, 609-735, Republic of Korea
bluepepe@pusan.ac.kr
[2] Department of Computer Science and Engineering, Pusan National University,
Busan, 609-735, Republic of Korea
hckwon@pusan.ac.kr

Abstract. A logistics information service manages a large amount of products and product transport flow. Many applications request logistics information from a logistics information service. For effective sharing of logistics information and knowledge, the design of a logistics information management system is important. The current web is changing to a semantic web that provides a common framework for data sharing. In this paper, we present a logistics information service architecture that supports a semantic web. Our logistics information service deals with RFID-sensed data and product-related data such as attribute, and containments. Logistics data is represented using the RDF for service to various applications.

1 Introduction

In logistics flow, according to the transportation of materials, a large amount data is transferred and shared. It is important to integrate and control a large amount logistics information according to the standard information management framework.

A warehouse or distribution center will receive the stock of a variety of products from suppliers and store these until receiving orders from customers. Within a wide logistics network, various data is shared and transferred among logistics subjects. Materials are stored in a warehouse or distribution center, and delivered to customers. Logistics automation systems can powerfully complement facilities provided by higher-level computer systems. A complete warehouse automation system can drastically reduce the workforce required to run a facility, with human input required only for a few tasks, such as choosing units of product from a bulk-packed case. Even here, assistance can be provided with equipment such as pick-to-light units. Smaller systems might only be required to handle parts of the process. In the flow of material through a network of transportation links and storage nodes, there is much logistics information generated by the automation system. To improve the efficiency of logistics operations, logistics automation is widely considered.

Recently Radio Frequency Identification (RFID) tags have been widely adapted to logistics, to the automatic identification of materials and to the tracking of containers.

Enterprise applications such as ERP and SCM integrate with logistics information services. An information integration and control system is important to provide overall control of the automation machinery and higher level functionality, such as identification of incoming deliveries, stock and scheduling of orders, and assignment of stock to outgoing trailers. In this paper, we present a logistics information service architecture based on Semantic Web for efficient managing and sharing of logistics information

2 Related Work

2.1 RFID

RFID technology uses wireless radio communications to quickly and easily identify individual products and items. It is one of the most promising and fastest growing automatic data collection technologies, opening new possibilities to improve business processes from manufacturing to supply chain management and beyond. Products can be identified uniquely and they can themselves communicate information for a wide range of business applications and solutions. In addition, RFID is more than just an ID code, since it can be used as a dynamic data carrier with information being written and updated to a label as a product moves along the product value chain [3].

The purpose of an RFID system [12] is to enable data to be transmitted by a portable device, called a tag, which is read by an RFID reader and processed according to the needs of a particular application. The data transmitted by the tag can provide identification or location information, or specifics attributes of the product tagged, such as price, color, date of purchase, and others.

RFID tags are often envisioned as a replacement for UPC or EAN bar-codes, having a number of important advantages over the older bar-code technology [3]. RFID codes are long enough that every RFID tag can have a unique code, whereas UPC codes are limited to a single code for all instances of a particular product. The uniqueness of RFID tags means that a product can be individually tracked as it moves from location to location.

An organization called EPCglobal is working on a proposed international standard for RFID and the Electronic Product Code (EPC) in the identification of any item in the supply chain for companies in any industry, anywhere in the world [3, 4].

2.2 Semantic Web

In the Semantic Web, an extension of the current web, information is given well-defined meaning, better enabling computers and people to work in cooperation [1]. The Semantic Web comprises and requires the following components in order to function: knowledge representation, ontologies, agents.

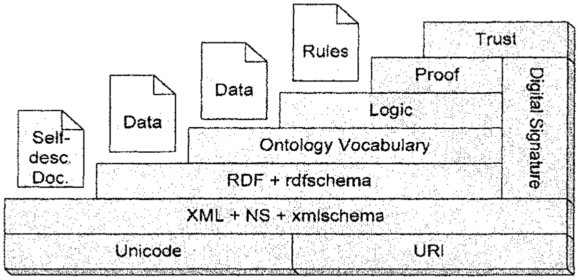


Figure 1. Semantic Web layered architecture [5]

The Semantic Web provides a common framework that allows data to be shared and reused across applications, enterprises, and community boundaries [5]. It is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework (RDF), which integrates a variety of applications using XML for syntax and URIs for naming

Recently, there has been much research about the efficient handling information of logistics information. W.S. Lo introduced a framework for the e-SCM multi-agent system, which combines ontology to improve flexibility of access with different terms [6]. There was also research on the ontology concepts for the SCM information infrastructure [7]. An approach to managing knowledge for coordination of e-business processes in the systematic application of semantic web technologies was introduced as semantic e-business [8]. Aabhas V Paliwal *et al.* proposed an OWL-S based approach for the automatic composition of Semantic Web Services [10].

3 Framework of RFID based Logistics Information Service

Logistics systems control the logistics flow that transports products from manufacturers to customers. In the process of product transport, many data related to logistics flows may be produced. RFID-based logistics systems create many more data. RFID-tagged data is some of data to be managed in logistics systems.

In the logistics environment, many applications require and exchange logistics information or knowledge about products. For the effective management of a large amount of logistics information such as product descriptions, transports of goods, and packing of products, logistics information management systems are required. In our research, the logistics information service managed a large amount of logistics information in providing information to related applications.

3.1 Logistics Information

There are many data in the logistics environment. A large amount of data is related to logistics flows. Logistics flow includes the transport steps of products such as manufacture, delivery, and use. It is important to control logistics flows in logistics systems

A logistics information service manages logistics information to control logistics flows and provide information on products. To control logistics flows, there should be sufficient information, and effective management of logistics information is also essential. Especially, RFID-based logistics systems handle much more information. Because RFID technology helps to recognize products automatically, a tremendous amount of RFID-related data is produced in the logistics flows. Also, to manage RFID-related data, much information such as product attributes, shipments, and containers, are required.

To establish the RFID-based logistics environment, the efficient handling of information on logistics is important. To handle logistics information efficiently, it is necessary to understand logistics flow and logistics information flow. RFID-based logistics information service handles four types of data: RFID-sensed data, attribute data, containment data, and transaction data.

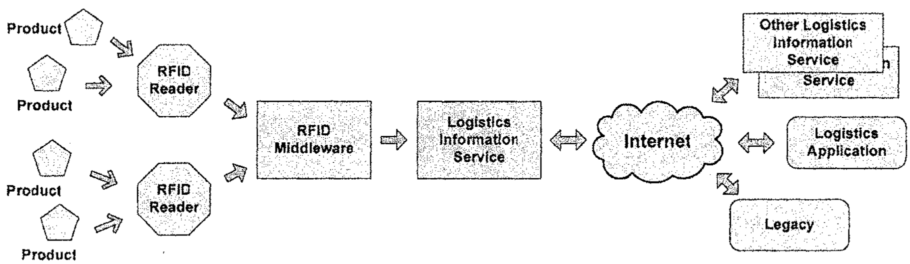


Figure 2. RFID-based logistics information flow

The RFID-sensed data is data that is automatically sensed by RFID technology. An RFID tag with an electronic code is attached to a product, and that product is identified by the electronic code in the RFID tag. When RFID-tagged products are delivered from manufacturers to customers, RFID readers sense the value written into the RFID tag of the product, as shown in *Figure 2.*

RFID-sensed data is collected and filtered by RFID middle-wares. A logistics information service stores RFID-sensed data received from RFID middle-wares. Also, a logistics information service provides this data to applications, legacy, and other logistics information services. RFID-sensed data basically consists of an electronic product code, the time that the RFID tag of products is read by an RFID reader, and the location of products that are detected by an RFID reader. Additional information such as temperature, humidity, the status of e-seals, etcetera, can be included in the RFID-sensed data.

Attribute data is data specifically on products. A company designs a model, makes prototypes of the model, and manufactures the products. Attributes of products are specifications of products and specific information about each product. Specifications of a product describe the common characteristics of all products grouped as one model. For example, model number, length, depth, ingredient, functions, and outward appearance are examples of specifications of products. Another type of attribute data is the attributes of each product. Each product has such attributes as product date, manufacturer name, manufactured factory, a certificate of origin, and others.

Containment data is data about the relationships between products and containing objects, or between contained objects and containing objects. Most products are packed into some containing objects, for instance, boxes, palettes, and containers. Also, packed products are loaded into trucks, trains, ships or airplanes. This data is also containment data. By the containment data, we can track where products are located.

Finally, transaction data is related to business transactions in logistics systems. Most products are transported from manufacturers to customers according to the contracts written between them. Transaction data is data related to contracts. Let us assume that a customer wants to buy some products from a manufacturer. After checking the specifications of a product, a customer orders some amount or number of that product. By this contract, the manufacturer transports the ordered amount or number of the product to the customer. To confirm whether the proper product and the proper amount or number of that product is transported, the relationships between contract documents and contracted products are set. Transaction data consists of contract identification and an electronic product code list.

3.2 Layers of logistics information service system

In the logistics system environment, the information service system plays important roles. The information service manages a large amount of logistics data and provides logistics information to applications. In RFID-based logistics systems, an especially large amount of logistics data has to be managed. For effective provision of logistics information, a logistics information service is designed on the basis of a web service. The logistics information service consists of four layers: the service layer, the data-handling layer, the data-access layer, and the repository layer.

The service layer consists of an interface for applications requesting logistics information and security modules. Authentication and authorization for accessing a logistics information service are in the security modules. Access to logistics information services is permitted only for an authorized user. An authorized user requests logistics information or provides information to the logistics information service. The Request Manager in the service layer controls requests for applications or other logistics information services. To relay messages between the logistics information service and applications, service layer uses SOAP (Simple Object Access Protocol).

The data-handling layer provides methods for managing the logistics information service. In other words, methods to control RFID-sensed data, attribute data, containment data, and transaction data are provided for application service requests. In the data-handling layers, there are four different data-handling modules, a service method manager, and an XML utility. The data-handling modules deal with each data types. For instance, the attribute data-handling module manages the attribute data of products, and the RFID-sensed data-handling module deals with data provided by RFID middleware. The methods of data-handling are controlled and published for applications by the service method manager. Methods are described using WSDL and are published. Legacy systems and applications can be made compatible with the methods of a logistics information service.

The data-access layer provides an accessing repository. In this layer, there is a repository accessing module that is commonly used by the data handling modules. The data-access module provides operations for storing and retrieving of logistics information in the repository.

Finally, the repository layer provides an XML repository. Logistics data is stored in XML format in this repository. XML data in the repository is managed by the data access module in the data access layer.

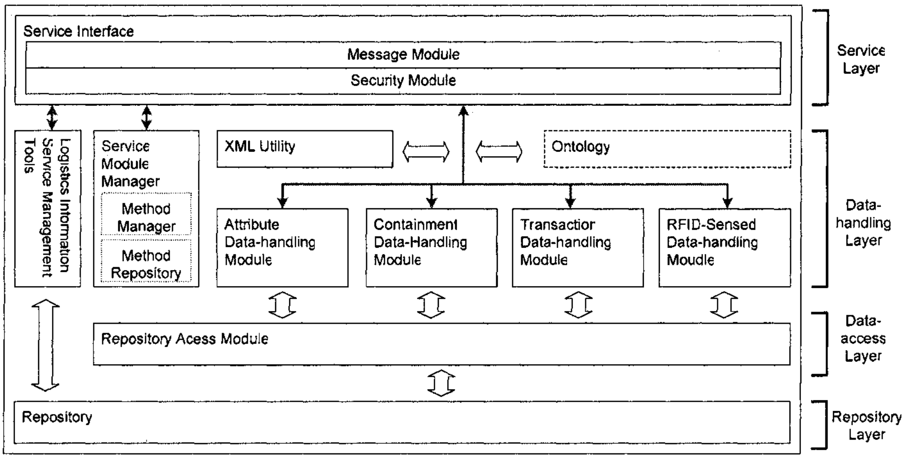


Figure 3. Logistics information service architecture

4 Semantic Web with logistics information service

A logistics information service is not limited to a local logistics system. A logistics information service provides logistics information to legacy systems or various logistics applications on the Internet. Any authorized users on the Internet can require logistics information. The current web is changing to a Semantic Web. A logistics information service should be applied to the Semantic Web.

4.1 Data Representation using RDF

A logistics information service should satisfy the following. First, it should be effectively store logistics data. Various types and large amounts of logistics data are stored for service. A logistics information service should support many requests of various applications such as ERP and SCM. Especially, effective sharing of logistics information with various applications is important. To effectively exchange information, we design logistics information service supporting the Semantic Web. Logistics information in the logistics information service is represented using ontology and RDF.

```
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:epc="http://www.pusan.ac.kr/ontology/epc#"
        xmlns:dc="http://purl.org/dc/elements/1.1/">
<epc:RFID_SensedData>
        <epc:ProductID>urn:epc:id:sgtin:15025.31.110</epc:ProductID>
        <epc:ReadDataTime>2005-06-10 11:34:50</epc:ReadDataTime>
        <epc:ReaderID>rd345612</epc:ReaderID>
        <epc:ReaderLocation>Jang-jeon, Busan</epc:ReaderLocation>
        <epc:ReaderType>Normal</epc:ReaderType>
</epc:RFID_SensedData>
</rdf:RDF>
```

Figure 4. An example of RFID-sensed data represented using RDF

4.2 Ontology module

To semantic processing of logistics information service, we set ontology module. Ontology module processes type checking of logistics data and constraint check.

4.2.1 Type conversion

There are various requests of logistics information from a legacy system or of logistics applications such as SCM and ERP. Applications can request information using their own data types, which are different from those of a logistics information service, even though their data types represent the same meaning. For sharing logistics information and knowledge, it is necessary to perform data-type conversion. Data-type conversion is controlled by the ontology of the data type. In this research, we focused on the ontology of time and of the measurement unit. Ontology improves interchangeability between legacy systems and a logistics information service. For example, a product manufactured at Location A is transported to Location B. Location A and B are in different time zones. If they both use local time, they cannot obtain exact time data. However, if they exchange time data that is converted to global time, they can obtain the right data. In logistics systems, there are many data related to data, time and unit of measurement. Ontology of date, time and measurement units can help to share of logistics information between a logistics information service and logistics applications.

4.2.2 Type and constraints check

In a logistics system, various types of products are transported by truck, ship, or train. There are many logistics data for transported products and transport flows in a logistics information service. In the transport processes, it is important to confirm the validity of product information and the validity of logistics flows. Our ontology module confirms the validity of product information such as expiration data, stock conditions, and logistics flow. Using ontology and RDF, this semantic processing is possible. For example, when a product in stock is sensed and the sensed data is inserted into the logistics information service, the ontology modules check whether the expiry date of the product is valid or not. The validity of the product is confirmed by the comparison of the expiry date of the product and the current date.

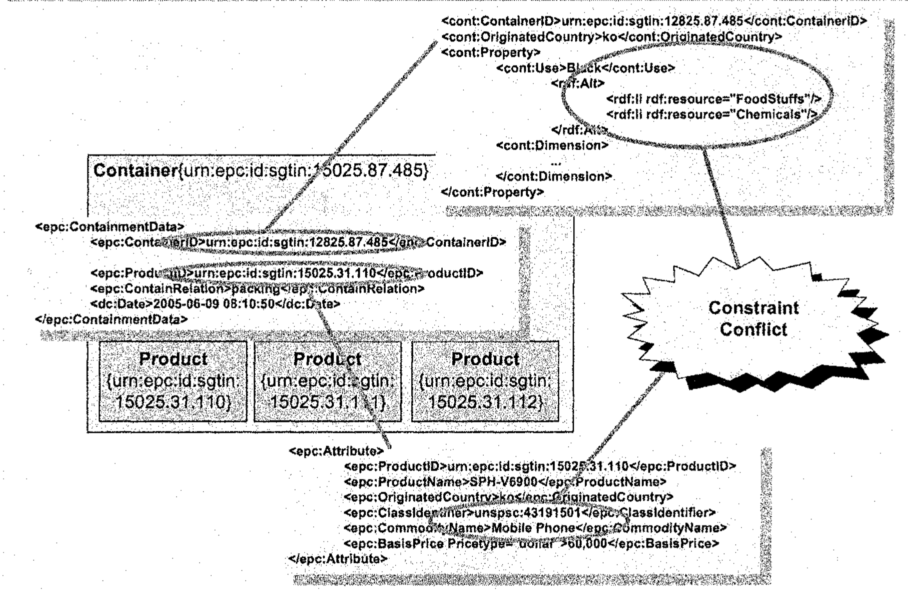


Figure 5. Constraints confirmation of logistics information

Figure 5 shows a constraint conflict of containment data. In the relationship between a container and products, even though the container cannot load the products owing to constraints, the products are loaded into the container. Logistics information service provides notification of constraint conflicts.

5 Conclusions and Future work

In this paper, we present RFID-based logistics information service architecture that manages logistics information: RFID-sensed data, product attributes, containment data, and transaction data. According to data type, we define different data-handling modules, and represented logistics information using RDF. Our logistics information service architecture is based on a web service for providing information to various applications. Also, to extend interoperability with applications, we used semantic web technology for our ontology module which provides flexibility of access and ensures the validity of logistics data.

Also in this paper, we designed the architecture of a logistic information service applying semantic web. In the future, we will implement this logistics information service and research effective semantic web service methods in a logistics system. And we will extend Ontology for the logistics information service.

Acknowledgement

This work was supported by Research Center for Logistics Information Technology (LIT) hosted by the Ministry of Education & Human Resources Development in Korea

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web, Scientific American, Vol. 284 (4). (2001) 34-43
2. A. Palival, N. Adam, C. Bornhovd, J. Sehaper: Semantic Discovery and Composition of Web Services for RFID Applications in Border Control, ISWC04, 2004
3. EPCglobal (http://www.epcglobalinc.org)
4. EPCglobal: The EPCglobal Network™: Overview of Design, Benefits, & Security, 2004
5. The World Wide Web Consortium (http://w3c.org)
6. Wei-shuo Lo, Tzung-Pei Hong, Shyue-Liang Wang, Yu-Hui Tao: Semantic web and Multiple-agents in SCM, International journal of Electronic Business Management, Vol. 2. (2004) 122-130
7. Inceon Paik, Wonhee Park: Software Component Architecture for an Information Infrastructure to Support Innovative Product Design in a Supply Chain, Journal of Organizational Computing and Electronic Commerce 15(2), (2005) 105-136
8. Michael N. Huhns, Larry M. Stephens, Nenad Ivezic: Automating Supply-Chain Management, Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems, (2002) 1017-1024
9. Albert Jones, Nenad Ivezic, Michael Gruninger : Toward Self-Integrating Software for Supply Chain Management : Information Systems Frontiers 3:4, (2001) 403-412
10. Rahul Singh, Lakshmi Iyer, A.F. Salam: Semantic eBusiness, Int'l Journal on Semantic Web & Information Systems, 1(1), (2005) 19-35
11. Chris Preist: A Conceptual Architecture for Semantic Web Services, Proceeding of International Semantic Web Conference, 2004
12. Sanjay E. Sarma, Stephen A. Weis, Daniel W. Engels.: RFID Systems and Security and Privacy Implications, In Workshop on Cryptographic Hardware and Embedded Systems, 454-470. LNCS, 2002

METRICS FOR OBJECTIVE ONTOLOGY EVALUATIONS

Robert J. Pefferly Jr.
Sunitia, Tallinn, Estonia, Tel:+372 5147 099, http://www.sunitia.com/
rob@sunitia.com

Michael C. Jaeger
TU Berlin, SEK FR6-10, Franklinstrasse 28/29, 10587 Berlin, Germany
mcj@cs.tu-berlin.de

Moussa Lo
UFR de Sciences Appliquees et de Technologie, Universite Gaston Berger, BP 234 Saint-Louis, Senegal, Tel:+221 961 2340, http://www.ugb.sn/
lom@ugb.sn

Abstract We present new metrics and techniques which allow one to configure a metadata catalogue and objectively describe knowledge management ontologies. Per C.E. Shannon (1948), when describing information based systems, statistical measures are a necessity; yet very few ontology based standards mention quantifiable measures such as entropy, data encapsulation, complexity, efficiency, evolution, or redundancy. We hope to demonstrate how statistical information measures can be implemented for ontology-based knowledge management systems using our $\mathbb{L}_0$ statistic, entropy, evolution, organization, sensitivity, and an interpretation of complexity.

1. Introduction

As demonstrated by Shannon, 1948, when evaluating information based systems, statistical measures are a necessity, yet unfortunately most popular management 'standards' that exist today rely too much on the subjective world of 'business processes' and not enough on consistent mathematical foundation principles. For examples, refer to Magkanaraki et al., 2000, Li et al., 2003, the Process Interchange Format, and Framework (PIF) and Workflow Management Coalition (WfMC) specifications. Thus, current knowledge management standards are inconsistent by definition and doomed to failure in large scale heterogeneous interactions. This emphasis on subjective 'business processes,' and lack of quantitative standards/measures has created a major stumbling block for the Business Process, Knowledge Management, Data Libraries, and Semantic Ontology fields.

Even though a great deal of effort has been spent on developing standards, as demonstrated in Magkanaraki et al., 2000 and the Santa Fe Institute work, even when objective measures are utilised, it is common to use 'hard number' evaluations when describing probabilistic 'soft number' informatic entities.

> For several reasons, the development of real-world enterprise-wide knowledge managements *ontology-based knowledge management systems* is still in the early stages. First, despite much research on ontology representations, engineering, and reasoning, features such as scalability, persistancy, reliability, and transactions - standardized and widely adopted in classical database-driven information systems - are typically not available in ontology-based systems. Maedche et al., 2003

The true innovation of this paper lies in the informatic metrics we prescribe for describing/comparing complex ontology based systems. We hope to demonstrate how objective statistical information measures can be implemented for objective knowledge management – one can now compare ontologies and answer the question: 'Is one ontology better than another?'

1.1 Background

Remark 1. *An ontology is a collection of symbols used to express data. Although ontologies possibly cover extremely technical fields or address complicated expertise, they are structurally nothing more complex than a finite set of symbols with a bijective mapping.*

In knowledge management systems, there are a number of 'big picture' issues that need to be explored, elaborated, and corrected since there is a disconnection between the popular 'Management Science' literature surrounding knowledge management and the underlying informatic principles. The emphasis on building 'Unambiguous Semantics' and 'Light-weight Inferences' such that ontologies have meaning to a human reader is a nice by product, but is immaterial and moot from an informatic perspective; it is the Shannon Information (i.e. Entropy $\mathcal{H}$) that is important. For example, whether we call a rose a 'roza,' 'roos,' 'fleur$_{rouge}$,' 'red.flower,' 'Mary's favourite,' 'Ω,' or 'H1026' - it is still a rose.

Remark 2. *For a practical example of an ontology, refer to the Information Ontology Root Li et al., 2003. Motik et al., 2002 presents a starting point for the practical questions: 'What is an ontology?' and 'How does one create an knowledge model?'*

There will always be variability and questionable assumptions when modeling physical problems, hence we are solely concerned with knowledge encapsulation to maximize the sufficiency of the metadata. In order to do this, not only does one have to capture a given degree of granularity, but one must also address how complex/efficient a system is using an respective ontology. Hence, entropy related measures provide the foundation for information metrics. Thus, the crux of any knowledge system is the information content and knowledge encapsulation that maximizes the sufficiency of the metadata; not in whether an ontology is subjectively 'linguistically correct' or follows 'common sense.'

> First, we realize that imposing a single ontology on the enterprise is difficult if not impossible. Maedche et al., 2003

Motik et al., 2002 attempts to put this into light with an objective view of ontology formulations, but once again forces systems to bend to a inconsistent world. Their argument on Light-weight Inferences is a representation of redundancy that aids a modeler in understanding the system such that she can envision how to make a system scalable and tractable. Organizations have not been able to break this a priori mindset as demonstrated by the standards that have been established; WFMC, PIF, and other ontology standards assume a general model that is built on business practices, this is fundamentally flawed. We argue that if the paradigm of the a priori data base is truly cast aside, then by using a Shannon Information approach, the system will be both scalable and tractable as well as efficient. This requires a complete point-of-view change from the programmer/developer that seems to be lacking in most ontology based designs - we continue to fight disorder and chaos instead of utilize it as a informatic tool.

1.2 Why are objective metrics necessary?

> ... the life cycle of ontology design can be summarised as three major stages, i.e. building, manipulating and maintaining. ... During developing the proposed system, establishing definite and consistent ontology is perhaps the toughest task. ... In this study, one year was spent to establish a common consensus and then to define the initial domain ontology in the metal industry. Li et al., 2003

This is not an uncommon occurrence where an 'inordinate amount of time' is spent on a 'repetitive and arduous task,' thus hindering development, management acceptance, and acceptability of knowledge management. Knowledge Engineers must justify this 'waste of time and resources' to management since current industrial standard ontologies are often not applicable to organizations without major modifications. A break from subjective evaluations to objective metrics would alleviate biased opinions and allow one to describe an ontology. Using appropriate metrics one can address building stage issues such as:

Building Stage:

- Instead of spending one year of a project developing an ontology, maybe only six months was substantially beneficial for effective information content. Points of diminishing returns can be identified and resources reallocated appropriately.
- One can finalize an iterative building process once the recommended changes add less than 'X' value to the knowledge management.

Manipulating Stage:

- The question of 'Is one ontology more effective than another?' can be ob jectively evaluated.
- How does one determine if the 'coverage' of an ontology is sufficient?
- What is the value added when changing elements in an ontology?

Maintaining Stage:

- Questions such as 'Is one ontology easier to maintain than another?' can be addressed.
- What does the addition or elimination of term 'X' do to the system? Will there be unintended consequences?

General: An iteration of domain expert solicitation must occur, but it is difficult to measure progress, hence, organizations must address issues such as:

- What is the value added with each iteration?
- When manipulating ontologies, how does one demonstrate positive improvement?
- When changing the terminology or structure of an ontology, how does one demonstrate the change is either beneficial, detrimental, or an exercise in futility?
- When maintaining an ontology, when does the redundancy and complexity become prohibitive?

1.3 Shannon discrete source model

Communication is simply a process of exchanging information via a channel encompassing { human - human, human - computer, computer - human, computer - computer } interactions It is hard to identify any information paradigm that does not follow a Shannon model where:

- Any phenomena which send discretized data via a finite set of packets is considered a Shannon discrete source and can be modeled via a Markov chain.
- The symbology of an ontology is secondary to the information that it encapsulates, thus, we are concerned with a series of symbols being used to represent data and the information that is passed via the symbolic chain.
- Whether data is transmitted as binary data packets, IP streams, words, smoke signals, or sounds, it is difficult to identify an ontology information paradigm that does not follow a Shannon model

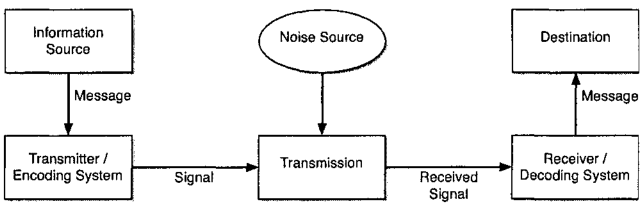


Figure 1. Shannon Model - Schematic diagram of a general communication system

2. Data, information, and metrics

Data is a hard number raw output of a discretized process represented by a countable set of symbols. Information is a level of abstraction above data, where data is a finite set of symbols such that the order of the symbols may contain 'meaning' - i.e. information as a sufficient statistic of a data set relative to a query.

> For the purposes of this chapter we use the term 'information' to refer to a spatio-temporal pattern that can be understood and described independently of its physical realization. Stephanie Forrest, 2000, page 362

Thus, one searches processed data to derive soft number information, where information is only 'meaningful' if it appropriately answers a query using an appropriate data set. The concept of data versus information is lost in most knowledge system literature; i.e. 'information' should be replaced with 'data.'

> Enterprises don't lack for **information**: they are drowning in it. So knowledge workers need all the help they can get in separating the wheat from the chaff. Savage, 2003

> Second, a large body of **information** in an enterprise typically already exists outside the knowledge manage system - for example, in other applications such as groupware, databases, and file systems. Motik et al., 2002

Definition 1. A hard number is a weighted singularity, such that it is a deterministic point estimate with 0 variance. Thus the expected value is $\mathbb{E}(\text{ Hard Number }) = \alpha$ and the variance is $\mathbb{V}(\text{ Hard Number }) = 0$.

Definition 2. A soft number is a stochastic point estimate where $\mathbb{E}(\text{ Soft Number }) = \alpha$ and $\mathbb{V}(\text{ Soft Number }) \neq 0$

Data is measured in hard numbers while soft numbers represent 'fuzzy' information values that do not have a tangible representation. For example, when looking at a picture, one does not individually observe every pixel (data) on a screen in strict numeric order to ascertain what the picture conveys (information). Thus, soft numbers are a sufficient description involving probability, expectations, and variances. Hence, when discussing hard numbers, $1 + 2 = 3$ without question. For soft numbers, $1 + 2 \neq 3$ is almost surely true, but given certain restrictions, $\mathbb{E}(1) + \mathbb{E}(2) = \mathbb{E}(3)$ may be true. Although the difference may seem as a 'techy' pet peeve, the implication for designing knowledge management systems are quite profound.

2.1 Complexity is not information

There a difference between statistical complexity and computational complexity and the common use of statistical complexity being referred to as a Shannon measurement equivalent to entropy is incorrect. Further, the common use of 'statistical complexity' being referred to as a Shannon measurement equivalent to entropy is incorrect. The following as an excellent example of the disconnection between complexity and entropy.

> It is emphasised that, given an entropy value, there are many possible complexity values, and vice versa; that is, the relationship between complexity and entropy is not one-to-one, but rather many-to-one or one-to-many. It is also emphasised that there are structure in the complexity-versus-entropy plots, and these structures depend on the details of a Markov chain or a regular language grammar. Li, 1991, page 381

2.2 New metrics

The following metrics were inspired by Hilderbrand, 1968 and further enhance the concepts presented in Martin Hepp, 2005, Section 2.
Definition 3. The computational complexity of a system F will be defined in terms of a non-standard. $\beta = \mathrm{O}(g(x))$, where β is 'Big-Oh' with respect to $g(x)$, when $\lim \frac{|\beta|}{g(x)} \leq K$ almost surely.

Computational complexity is sometimes referred to as Kolmogorov complexity, Gacs, 2001 but we will utilize a non-standard Big-Oh notation where the magnitude of K will be of importance.

Definition 4. The $\mathbb{L}_0$ statistical complexity is defined by the inverse of the following expectation where $\mathtt{card}\,(\mho)$ is the cardinality of the domain space:

$$\mathbb{L}_0\left(F_N\left(x\right)\right) = \mathbb{E}\left[\sqrt{\frac{\mathbb{P}\left(x\right)}{\mathtt{card}\left(\mho\right)}}\right]^{-1} = \frac{1}{\sum\limits_{j=1}^{N} \mathbb{P}\left(x_j\right)\sqrt{\frac{\mathbb{P}(x_j)}{\mathtt{card}(\mho)}}} \tag{1}$$

As discussed in Shannon, 1948 a uniform distribution is optimal for information content, thus our $\mathbb{L}_0$ statistic is based on the Discrete Uniform Probability Distribution and has a lower bound of 0 with an unlimited upper bound. For example, an equiprobably coin has a $\mathbb{L}_0$ value of 2 and a 6 sided equiprobably die has a $\mathbb{L}_0$ value of 6.

Definition 5. The organization or structure of a process will be

$$\Phi\left(F_N\left(x\right)\right) = \mathbb{E}\left(\mathbb{P}\left(x\right)\right) = \sum_{j=1}^{N} \mathbb{P}^2\left(x_j\right) \tag{2}$$

There is a definite need to distinguish between organization and complexity but most authors assume a limited view on how 'correlation typically provides a lower bound of a measure of complexity.' Organized structures are independent of both the entropy and complexity, thus a separate metric Φ, is a necessity.

Definition 6. The evolution of a process will be $\mathcal{E}\left(F\left(x\right)\right) = \Phi\left(F\left(x\right)\right)^{-1}$.
A simple structure indicates that a system is highly organized and the more organized a system is, the smaller its evolution. The difference between two ontology evolution values represent a measure of informatic distance. To draw an analogy: physical mass is measured in grams and distance in meters; information 'mass' is measured in $\mathbb{L}_0$ units and the distance is measured using evolution.

Definition 7. The sensitivity is a first order difference (change) in the evolution of a system when an element is either eliminated (or added) from the process. Such that $\mathcal{S}\left(F_N\left(x\right)\right) = \Delta\mathcal{E}\left(F_N\left(x\right)\right) = \mathcal{E}\left(F_{N\pm1}\left(x\right)\right) - \mathcal{E}\left(F_N\left(x\right)\right)$, where $\mathtt{card}\left(F_N\right) = N$ and $\mathtt{card}\left(F_{N\pm1}\right) = N \pm 1$.

Sensitivity will be a measure similar to the sensitivity of a numerical approximation, in that it will be a measure of the effect a small change in the structure has on the overall system. For our purposes, sensitivity will be measured as a finite difference of the organization metric via the addition/deletion of a term.

2.3 Standard Metrics

The following metrics are standard definitions as defined by Shannon, 1948:

Definition 8. The entropy $\mathcal{H}$ of a system will be defined as:

$$\mathcal{H}\left(F_N\left(x\right)\right) = \mathbb{E}\left(-log\left(\mathbb{P}\left(x\right)\right)\right) = -C\sum_{j=1}^{N}\mathbb{P}\left(x_j\right)log\left(\mathbb{P}\left(x_j\right)\right) \qquad (3)$$

per Shannon, 1948, Theorem 2. The choice of C is merely a normalizing correction for the a unit measurement.

Notation 9. Relative entropy is the percentage of entropy realized with a coding system as compared to the Shannon theoretic value for the maximal entropy of a system. The relative entropy of a system F will be defined as $\mathcal{H}_{Rel}\left(F_N\right) = \frac{\mathcal{H}_{practical}(F_N)}{\mathcal{H}_{theoretical}(F_N)}$.

Relative entropy is the percentage of entropy realized with a coding system as compared to the Shannon theoretic value for the maximal entropy of a system.

Notation 10. The redundancy of a system F will be defined as $\mathcal{R} = 1 - \mathcal{H}_{rel}\left(F_N\right)$.

2.4 Practical Examples

The following Table 1 indicates the relative metrics for the examples of this section. The sensitivity is calculated by removing the first element.

Metric	Theoretical die	Bayesian die	A priori ontology	Bayesian ontology
$\mathcal{H}$	0.778	0.736	0.7009	0.714
$\mathcal{H}_{Rel}$	1	0.946	0.9007	0.9176
$\mathcal{R}$	0	0.0536	0.0993	0.0824
O	6	6	6	6
$\mathbb{L}_0$	6	5.592	5.334	5.442
Φ	0.1667	0.2	0.22	0.2099
$\mathcal{E}$	6	5	4.545	4.765
$\mathcal{S}$	-1	-0.736	-0.396	-0.451

Table 1. Metric results

A six sided die - Theoretical. For a relatively easy example, assume that one is trying to describe the physical phenomena of rolling a six sided equiprobable die. The die operates in a discrete space where each side of the die has a one in six chance of appearing, therefore, the probability density function (PDF) $\mathfrak{f} \sim \mathcal{U}\left(\frac{1}{6}\right)$, $\mathbb{E}(\alpha) = 3\frac{1}{2}$, $\mathbb{V}(\alpha) = 2\frac{11}{12}$. Notice that this is information, how often does one role a six sided die and end up with $3\frac{1}{2}$?

A six sided die - Bayesian. Assuming that the six sided die is 'too complex' for one to understand, roll the die $N = 10$ times and plot the histogram of the data. Assuming that a six sided die $\widehat{F}$ is rolled, the realization of a the result of the die will consist of an iid random data set is $\{3, 1, 6, 6, 4, 5, 2, 3, 6, 5\}$. Using this information one can estimate that: $\widehat{\mu}_\alpha = 4.1$ and $\widehat{\sigma}^2_\alpha = 3.211$.

Theoretical Ontology. A six term ontology is designed by a knowledge engineer who estimates a priori probability values for filling data elements per values outlined in Table 2.

Element (x)	$\mathbb{P}(x)$
Version	$\frac{1}{20}$
Summary	$\frac{1}{20}$
Organization	$\frac{3}{20}$
Date	$\frac{4}{20}$
Author	$\frac{5}{20}$
Title	$\frac{6}{20}$

Table 2. A priori ontology

Practical Ontology. The same ontology is utilized where Bayesian a posterior probabilities are derived from 'real instances' stored in the catalogue; as shown in Table 3. In this table, five instances of a model are utilized

Figure 2. PDF versus histogram

$M1, M2, M3, M4, M5$ where an X implies that the data entry is filled for that instance.

Element (x)	M1	M2	M3	M4	M5	$\mathbb{P}(x)$
Version		X				$\frac{1}{18}$
Summary				X		$\frac{1}{18}$
Organization	X	X		X		$\frac{3}{18}$
Date	X	X	X	X		$\frac{4}{18}$
Author	X		X	X	X	$\frac{4}{18}$
Title	X	X	X	X	X	$\frac{5}{18}$

Table 3. A posterior ontology

3. Conclusion

In this paper we postulate that there is a distinct disconnection between complexity, entropy, and the structure/organization of an ontology, where a lack of objective metrics is the crux of the knowledge management problem. We hope that others will question our 'new' measures and independently vindicate or vilify our constructs. We stress that our metrics warrant independent verification.

Ontology management is not a trivial matter since knowledge management users delete/add terminology, data elements and their inter-dependencies change

> Changing an ontology can induce inconsistencies in other parts of the ontology. Semantic inconsistency arises if an ontology entity's meaning changes. (...) An ontology update might also corrupt ontologies that depend on the modified ontology and, consequently, all artefacts based on these ontologies. (...) However, apart from syntax inconsistency, semantic inconsistency can also arise when, for example, the dependent ontology already contains a concept that is added to the original ontology. Maedche et al., 2003

Hence, there is an justifiable need for objective metrics such that different knowledge management ontologies can be compared on equal footing. This is still a very subjective field when objectivity should be the goal of the discipline in order to remove the subjective nature surrounding knowledge, thus minimizing miscommunication. By establishing objective measures implementing metadata metrics, solid definitions will improve the overall state of the knowledge management field - if you can measure it, you can manage it.

References

Stephanie Forrest, Steven A. Hofmeyr, *Immunology as Information Processing*, Oxford University Press, England, 2000, Eds. L.A. Segel and I.R. Cohen, Design Principles for Immune System & Other Distributed Autonomous Systems, Peter Gacs, *Quantum Algorithmic Entropy*, J. Phys. A: Math. Gen. 34 (2001), 6859-6880, June 2001

Martin Hepp, Joerg Leukel, Volker Schmitz, *Content Metrics for Products and Services Categorization Standards*, Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, pages 740-745, April 2005

Francis B. Hilderbrand, *Finite-Difference Equations and Simulations*, Prentice-Hall, Englewood Cliffs (NJ), 1968

Sheng-Tun Li and Huang-Chih Hsieh and I-Wei Sun, *An Ontology-based Knowledge Management System for the Metal Industry*, The Twelfth International World Wide Web Conference (WWW'03), Budapest, Hungary, May 2003

Alexander Maedche and Boris Motik and Lijiljana Stojanovic and Rudi Struder and Raphael Volz, *Ontologies for Enterprise Knowledge Management*, IEEE Intelighent Systems, pages 26-33, Volume 18, Number 2, 2003

Aimilia Magkanaraki and Sofia Alexaki and Vassilis Christophides and Dimitris Plexousakis, *Benchmarking RDF Schemas for the Semantic Web*, Institute of Computer Science, FORTH, Heraklion, Greece, EU Project OntoWeb and Question-How, Volume 6, 2000

Boris Motik and Alexander Maedche and Raphael Volz, *A Conceptual Modeling Approach for Semantis-Driven Enterpise Applications*, Springer - (ODBASE-2002), California http://kaon.semanticweb.org/papers, 2002

Peter R. Savage, *The Data Alchemists*, IEEE Spectrum Online, July 2003

Claude E. Shannon, *A Mathematical Theory of Communication*, The Bell System Technical Journal, pages 379-423, Volume 17, No. 3, 1948

Wentian Li, *On the Relationship Between Complexity and Entropy for Markov Chains and Regular Languages*, Complex Systems, pages 381-399, Volume 5, No. 4, 1991

M-ADVANTAGE

Multimedia - Automatic Digital Video & Audio Network Through Advanced Publishing European Service

Andrea de Polo on the behalf of the M-ADVANTAGE Consortium
Affiliation: Fratelli Alinari Photo Archive, Florence, Italy

Abstract: The research and development efforts within the M-ADVANTAGE project described in this paper aim at increasing the competitiveness of Europe's digital content industries by semantic-based services across the content value chain, including personalized delivery. Nowadays, user needs are addressed by costly solutions that require intensive human intervention. The described activities strive at filling the gap in automatic processing of multimedia by creating an intelligent infrastructure allowing considerable productivity gains. For achieving this goal, it is proposed to carry out a tightly integrated research and development activities also in terms of the blend of research, technology, content and user partners involved.

The research and development targets to build a service infrastructure for automated semantic discovery, extraction, summarization, labelling, composition, and personalized delivery of content from heterogeneous multimedia repositories. This will involve foundational research such as data models and ontologies required for merging multiple heterogeneous data types into an integral representation; component-level research for parts of the service infrastructure; as well as development of semantic-based productivity tools. Making use of established Semantic Web, multimedia description and other standards are anticipated to enable a broad uptake of M-ADVANTAGE's open source and non proprietary technologies.

While the project's research & technology partners include leading university centres and industry players (including SMEs), on the content side renowned private as well as public organisations are involved. They hold and provide access to all types of content such as audiovisual, stock image archives, news agency and other content, and strive to develop and market knowledge-based content services.

Key words: service portability, service adaptation, network interoperability, context awareness, semantic ontology, industrial Semantic Web environment

1. STATE OF THE ART

M-ADVANTAGE, being an integrated project, has a wealth of tasks, processes and scientific and technological objectives. Due to the nature of the work, these can be grouped under the umbrella of intelligent multimedia analysis and access with the use of ontological information. Thus, the most relevant state of the art is that related to the development of ontological knowledge representations for multimedia applications as well as those related to multimedia analysis and access approaches.

As far as representation is concerned, the MPEG-7 standard, formally named "Multimedia Content Description Interface", provides a rich set of standardized tools to describe multimedia content. However, in order to make MPEG-7 accessible, re-usable and interoperable with many domains, the semantics of the MPEG-7 metadata terms need to be expressed in an ontology using a machine-understandable language. Additionally, there is an increasing need to allow some degree of machine interpretation of multimedia information's meaning. To this end, several approaches in the literature address the problem of building multimedia ontologies to enable the inclusion and exchange of multimedia content through a common understanding of the multimedia content description and semantic information.

Hunter [1] describes the trials and tribulations of building a multimedia ontology represented in RDF Schema [2] and demonstrates how this ontology can be exploited and reused by other communities on the semantic web (such as TV-Anytime [3], MPEG-21 [4], NewsML [5], museum, educational and geospatial domains). A core subset of the XML-based MPEG-7 specifications together with a top-down approach to generate the ontology is used. The first step is to determine the basic multimedia entities (classes) and their hierarchies from the MPEG-7 Multimedia Description Scheme (MDS) basic entities [6]. The RDF schema semantic definitions for MPEG-7 can be linked to their corresponding pre-existing MPEG-7 XML schema definitions. Additionally, the RDF Schema can be merged with RDF schemas from other domains to generate a single "super-ontology" called MetaNet. Expressed in DAML+OIL [7], MetaNet can be used to provide common semantic understanding between domains. This super-ontology can be used to enable the co-existence of interoperability, extensibility and diversity within metadata descriptions generated by integrating metadata terms from different domains.

The proposed method for building a multimedia ontology has been applied to manage the manufacturing, performance and image data captured from fuel cell components [8,9]. Future work plan of [1] includes the automatic semantic extraction from the MPEG-7 XML schema document as well as the automatic linking of the semantics to the XML schema document. Acknowledging the importance of coupling domain-specific and low-level description vocabularies, a similar methodology for enabling interoperability of OWL domain-specific ontologies with the complete MPEG-7 MDS is described in [10]. The approach is based on an OWL ontology, referred as a core ontology, which fully captures the MPEG-7 MDS. For the development of the core ontology, a set of rules is defined to map particular MPEG-7 components to OWL statements. The integration of the domain-specific knowledge is performed by considering the domain-specific ontologies as comprising the second layer of the semantic metadata model used in the DS-MIRF framework (the first layer is encapsulated in the so called core ontology). For this reason a set of methodological steps is provided. Additionally, rules are provided for transforming the OWL/RDF metadata, structured according to the core ontology and the domain-specific ontologies, into MPEG-7 compliant metadata. Following this approach proves advantageous for MPEG-7-based multimedia content services, such as search and filtering services, since incorporating semantics can lead to more accurate and meaningful results in terms of meeting the user queries.

The most conventional approach is the keyword approach, which is based solely on textual metadata annotation. Such search technologies often exacerbate information overload; although they can identify documents in which a search term appears, they cannot tell how relevant the document is to the subject being researched. They simply look for the occurrence of keywords and are unable to decipher whether the concept represented by a search term is related to the main idea of a document. This approach follows closely the developments in the field of simple text retrieval [11], which has not progressed much since 1999.

On the other hand, semantic indexing aims at finding "patterns" in unstructured data (documents without descriptors such as keywords or special tags) and use these patterns to offer more effective search and categorization services [12]. Semantic indexing techniques are language-agnostic, so data collections do not have to be in English, or even in any specific language. This approach comes closer to bridging the semantic gap, seen as the discrepancy between the capabilities of a machine and that of a human to perceive visual content. In order to understand the content of an image, one necessary step is to identify objects within it. The aim is not absolute recognition of each individual object in the image but to enable similarity search on image parts immersed into various contexts. A possible route to achieve content understanding is the direct and automated extraction of textual descriptors from visual content directly.

In most auto-annotation efforts, prediction is done at keyword (or concept) level and each concept is predicted independently from the other. It is therefore possible to obtain incoherent predictions such as "space" and "indoor" simultaneously when describing the same image. There is need for a global maintenance of semantic coherence between parts of the annotation. This clearly requires the use of a consistent and normalized multimedia description scheme, which will be defined as a formal structure of digital meta-publication, where digital meta-publication means a set of connected digital objects (text, audio, video, etc.) with a strict hierarchy, advanced metadata information and other sophisticated possibilities.

When it comes to retrieval, semantic multimedia retrieval requires the presence of an already annotated multimedia content. There are several types of semantic retrieval, all of which utilize semantic matching algorithms between the semantic content descriptions. The first type of semantic retrieval is based on direct description / definition of the Semantic Track of the target data by a user. Such process of semantic definition requires a user-friendly interface with features for Ontology browsing. Precise results may be retrieved via specifying the significance of semantic features. A second type of semantic retrieval is based on describing / defining the Semantic Track of the target data by a user indirectly through defining the initial similar multimedia data. Thus, a sample of media file with a set of extracted mathematical features is used as an input query. Precise results may be retrieved via specifying the significance of mathematical features. The third type represents a kind of combination of the previous two semantic retrieval types.

Using a combination of Bayesian Inference and Signal Processing Technology (SPT, Shannon's Information theory), can indeed help in the automatic extraction of key conceptual aspects of any piece of unstructured information (documents, web pages, emails, voice, videos, images ,etc). Bayesian Inference is a mathematical technique for modelling the significance of semantic concepts (ideas) based on how they occur in conjunction with other concepts. By applying contemporary computational power to the concepts pioneered by Bayes, it is now feasible to calculate the relationships between many variables quickly and efficiently, allowing software to manipulate concepts.

Information Theory provides a mechanism for being able to extract the most meaningful ideas in documents, thus leading us to the definition of a "pattern matching" technology. Information Theory is the mathematical foundation for all digital communication systems. Natural languages contain a high degree of redundancy. A conversation in a noisy room can be understood even when some of the words cannot be heard; the essence of a news article can be obtained by skimming over the text. Information theory provides a framework for extracting the concepts from redundancy. Shannon's theory is that "the less frequently a unit of communication occurs, the more information in conveys". Therefore, ideas, which are rarer within the context of a communication, tend to be more indicative of its meaning.

The Pattern-matching approach has the additional benefits: (a) it is robust to false positive matches and (b) it can determine how similar documents are, without both documents being tagged the same way, or even tagged at all; this is called idea distancing.

In the next sections of the paper we will provide and in depth description of the technical and scientific solution that we would like to achieve within the project.

2. SCIENTIFIC AND TECHNICAL OBJECTIVES

The M-ADVANTAGE project aims at developing an infrastructure capable of delivering multimedia information and content customized to the needs of end-users. It focuses on building some specific components to provide the functionalities necessary to facilitate the construction of advanced multimedia content applications and the use of structured and unstructured multimedia information.

The goal of the M-ADVANTAGE approach to the "delivering multimedia information and content customized to the needs of end-users" is based on three ambitious deliverables:

- M-ADVANTAGE is able to automatically integrate heterogeneous multimedia content. Since the integration is automatic as a result the M-ADVANTAGE infrastructure is highly scalable and will be able to expand the current 6 content flows to an unlimited number, simply by adapting the hardware infrastructure, accordingly.
- 360° Technology Approach: M-ADVANTAGE infrastructure is based on the more up-to-date technology approaches for managing unstructured information: Keyword, Semantic and Statistical (through a pattern matching system).
- Develop specific application services to deliver the content managed by the M-ADVANTAGE back-end infrastructure

These features will enable the utilization of digital content delivery systems distributed across the computer network and will process the information stored within these archives in order to find dependencies, links and similarities between various pieces of information. This will allow to automatically manage and customize the available content for the needs of end-user applications built on top of the M-ADVANTAGE infrastructure.

From the scientific point of view, the following contributions are expected from the M-ADVANTAGE project to the research community:

- Automated multimedia (semantic) discovery, which concerns both retrieval, i.e. search for multimedia files; and extraction, i.e. more focused search for specific structural components of the multimedia: episodes, frames, images (focuses), etc.
- Advanced video summarization, i.e. content of the whole video clip can be browsed quickly.
- Advanced techniques for semantic labelling, i.e. propagation of labels through hierarchical database structures.
- Automated multimedia integration / composition: real power is in composition of different structural elements (episodes, frames, focuses) extracted from heterogeneous multimedia files in a coherent track.
- Semantic personalized delivery: based on semantic interactions of user activities / actions on content and user's explicit preferences; proactive supply to the user of relevant multimedia.
- Interoperability between heterogeneous (web-) services and multimedia: this is possible following Semantic Web's recommendations about common (upper-) ontology or managing mapping between semantic concepts from different ontologies.

From the technical point of view, the M-ADVANTAGE platform aims at creating a state-of-the-art cutting edge technology that is going to serve public and business sector in the Knowledge Management for multimedia content. In this respect:

- Statistical search will be used as a super set of the conventional methods in the sense that where these should fail it will always be possible with this methodology to grasp concepts embedded in images, text and videos together and deliver a complete content overview on somebody's search.
- The combination of semantic / ontology methodologies and the statistical one will offer users the possibility to have a much more precise and to the point interaction with the KB. Users will be profiled and grouped into communities according to their previous interactions with the KB.
- Content will be classified automatically according to concepts that undoubtedly identify it. It will also be possible to split any video content into it's fundamental scenes using "scene detection" and "object extraction" techniques, thus allowing editors the possibility to reassemble a piece of video according to their needs. It will also be possible to search on the text extracted from a speech in a video using the most technological

approach of speech to text technology, and returning the meaningful frames that relate to the search argument. All of these tasks shall be carried out automatically without the user noticing it.

Overall, the goal of the M-ADVANTAGE platform is to offer users a multimedia access experience that combines all that is needed to one time visitors and to professional users.

3. OUTLINE IMPLEMENTATION PLAN

M-ADVANTAGE platform intends to provide an integrated solution for the B2B value chain starting from the content owners (image archives, public domain digital libraries, multimedia online deposits, etc), passing through the added-value content creators (press agencies, publishers, creative sector, etc) and arriving to the service providers (Internet portals, broadcasters, news, etc).

This can be broken down into the segmentations described in Figure 1 representing an in depth view of the content value chain that M-ADVANTAGE intends to address.

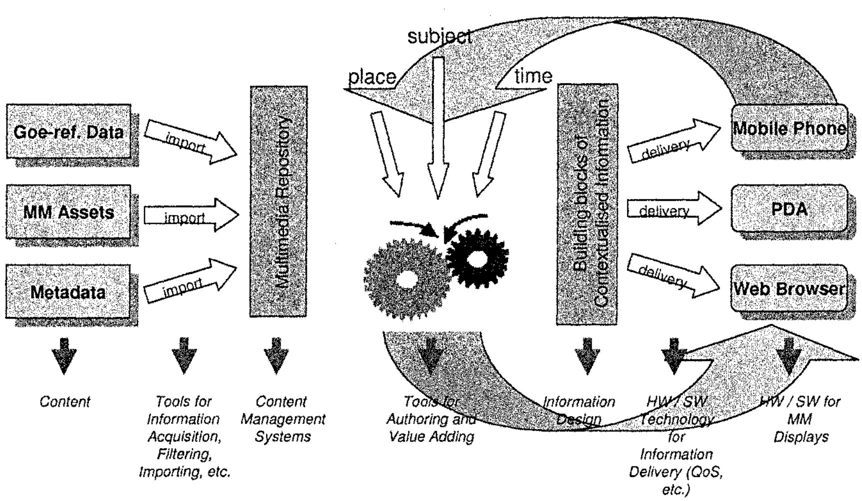


Figure 1. M-ADVANTAGE services from a technical point of view. In this sketch we see from the left to the right the workflow of the project, from input (content providers) to output (new services, enriched content and so on).

From the point of view of the users, on the other hand, what M-ADVANTAGE mainly contributes is the consideration of knowledge in the process of access to multimedia content, as described in Figure 2. The integrated wizard helps the online editor or content creator to enrich its valuable multimedia material with innovative and unique features.

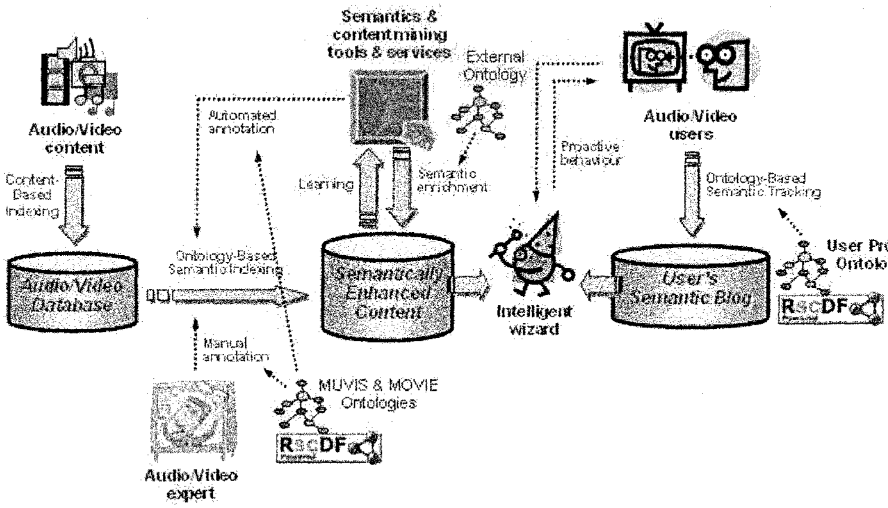


Figure 2. M-ADVANTAGE services from the users' point of view

M-ADVANTAGE platform is targeted to improve the industrial processes of the various actors involved in the generation of multimedia information services, by:

- Providing access to a larger amount on multimedia information,
- Simplifying search activities,
- Offering an integrated Digital Rights Management System (DRMS)
- Providing a Customized Multi Licensee Service for different level of subscribers, partners and end users with different kind of features and permissions available, according to the subscription level
- Offering a secure online payment mechanism, including a strong mechanism to guarantee the user's privacy
- Pay per View
- Automatic tasks to enrich "poor" or unclassified items with enriched multimedia features
- Offering customized and personalized search environments, using, for example:
    - Online wizard: product creation,

- Smart on line assistant: 3D tutorial aid character,
- software tracking for user behaviour

The M-ADVANTAGE platform is intended to be a basis for a wide set of tools. These tools will satisfy the different needs of the different actors involved, taking in consideration their different business approaches (profit private organization V/S public non-profit bodies), their different technological situation (on-line, fully digital, advanced business V/S traditional archives with limited informatics support), their different vocations (highest protection of rights V/S widest accessibility to contents).

M-ADVANTAGE aims to research, develop, implement, integrate, and test with users, the enabling technologies required to realize the M-ADVANTAGE concept. In order for all these aspects, as seen from the users' and technical points of view, to be supported, M-ADVANTAGE needs to integrate and/or develop a wide range of tools and services, as briefly summarized in Figure 3. As most of them (one may consider automatic indexing of any type of media as an example) are complex services the implementation of which remains in some cases an open research issue, the work is organized in a set of work packages, some of which have a research oriented nature, rather than a pure development and integrating one.

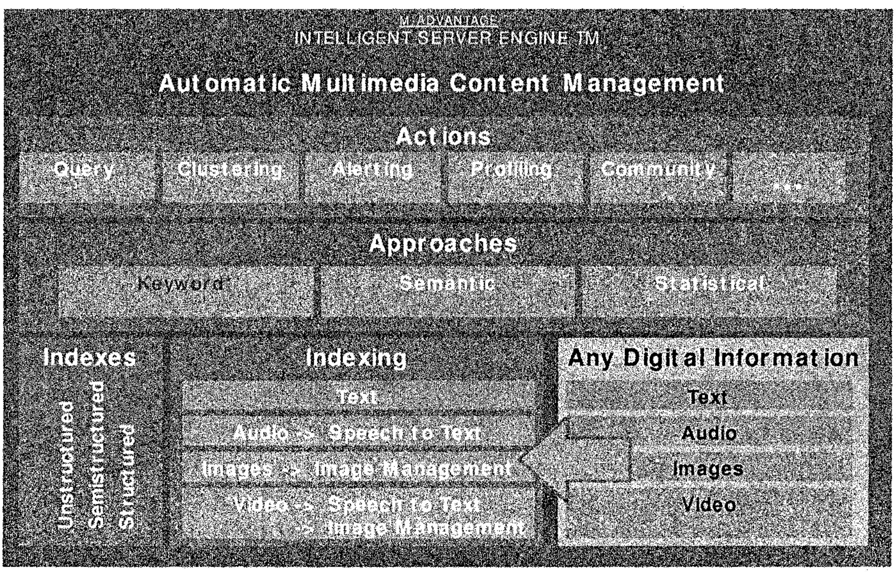


Figure 3. Business services included in M-ADVANTAGE available for the members of the system

Figure 4 below shows the overall architecture proposed for the M-ADVANTAGE platform, and the way in which the results of the various work packages come together in order to meet the projects objectives. In the following sections we will explain how the architecture's components (A-D) will be developed.

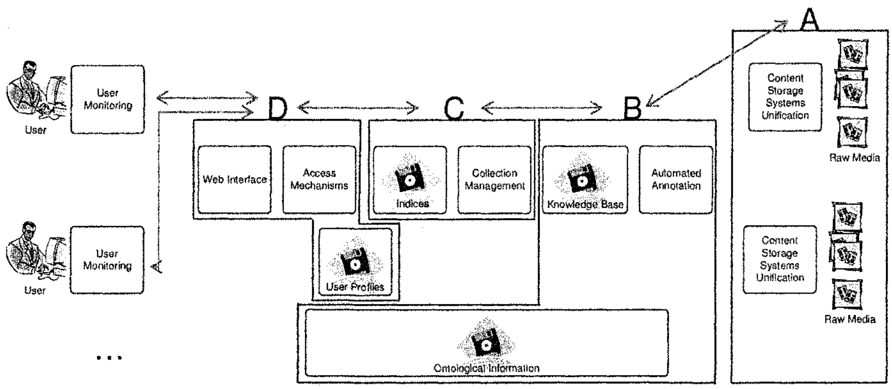


Figure 4. M-ADVANTAGE architecture and its related workflow.

Development of component, service 1

One of the main objectives of the concerned research is to make possible the access and consideration of heterogeneous archives, in order to provide a platform able to unify the wealth and diversity of archives currently existing and/or under development. Thus, the first service considered in the architectural design of the platform is that of the analysis of existing content storage systems and the specification of a generic querying and access interface, able to support and serve all existing content. In this process the needs of subsequent services will also be considered in order to make sure that no constraints on the ability to provide efficient and intelligent services is generated as an artifact. Based on this generic interface, it will be possible to create software interfaces, custom to each archive, that will allow for the automatic connection of the archive with the overall M-ADVANTAGE system.

Development of component, service 2

The most important and challenging objective of M-ADVANTAGE is to contribute to the effort to bridge the semantic gap by following innovative, knowledge based approaches to semi-automatic and fully automatic media annotation. For this purpose, complex ontological data models will be developed, and organized in Mathematical feature ontology and Mathematical feature ontology, as well as a reasoning framework capable to consider and reason with such input. This infrastructure will be used in order to automatically analyze the content acquired from the various media archives and generate the knowledge base. Various methodologies will be supported in this process, ranging from simple manual annotation to semi-automatic, retainable and adaptive computer assisted annotation and fully automated knowledge driven annotation.

Specifically for the manual annotation approach, the concept of visual representation of content is an important element and it is known that high data dimensionality (as that typically defined in multimedia processing) hinders its visualization and hence its handling. M-ADVANTAGE will perform data-adaptive dimension reduction, as opposed to classical homogeneous dimension reduction. The above structural characterization will guide dimension grouping and selection to arrive to a data representation suitable for visualization and interaction.

Development of component, service 3

The meta-publication description format used to store the analysis results in the knowledge base will also be designed in way that provides for optimal balance between effectiveness (in terms of descriptive power) and efficiency (in terms of space and processing power) for storage and consequent processes.

Clearly, M-ADVANTAGE is not designed to serve a single or a mall number of archives. Thus, characterizing the diversity of the multimedia collection at hand will be essential for obtaining a reliable cue of the available multimedia space. Typically, measuring such a parameter includes locating dense and sparse parts of the feature space, as populated by the multimedia collection. Several studies in the fields of data mining and related provide efficient tools for doing so. In M-ADVANTAGE we will also experiment with radically different approaches based on discrete global optimization procedures inherited from the field of Operations Research. We assert that not only this field is providing tools for operating transforms on the multimedia collection, as required by our context but also helps in designing gauging tools that will provide measures on the opaque collection to enhance subsequent operations.

While the above provides fundamental measurements of the collection properties and structures, M-ADVANTAGE will also address the issue in concrete terms of clustering and informative sampling, underlying the classical tasks of filtering and summarizing multimedia information collections. Again, solutions defined will directly apply to concrete issues such as that of proposing interactive multimedia catalogues of document collections. In particular, the aim is to instantiate the concept of collection guiding that extends classical browsing by creating exploration strategies around the document collection and therefore literally guide the user through it.

The result of the abovementioned work will be the generation of information rich indices, ideal for the subsequent access procedures.

Development of component, service 4

Finally, M-ADVANTAGE aims to offer state of the art and innovative, intelligent, personalized multimedia search and access services to end users. In this respect, a state of the art content management system will be integrated in the overall platform, allowing for simple (keyword based), semantic and statistical search in the available indices; in all of these search approaches, knowledge contained in the ontological databases will also be considered. Additionally, user interactions with the M-ADVANTAGE platform will be logged and analyzed in order to extract user profiles that can be fed back into the system thus enhancing the quality of services offered to each specific user; the representation of user profiles will be based on a properly designed profile ontology, thus allowing for the optimal consideration of available ontological information in the extraction, as well as in the utilization, of the user profiles. Finally, additionally to the conventional access through web interfaces, a tool capable of operating totally unsupervised, monitoring local user activity, utilizing it to extract preference information and query for interesting documents will be integrated in the platform. M-ADVANTAGE knowledge bases will be accessible not only via the browser or the M-Assistant but also via a virtual assistant (such as on www.alfaromeo.it or www.agenttalk.nl).

4. CONCLUSIONS

The M-ADVANTAGE project aims to deliver a first version of an infrastructure capable of delivering multimedia information and content customized to the needs of end-users. It focuses on building some specific components to provide the functionalities necessary to facilitate the construction of advanced multimedia content applications and the use of structured and unstructured multimedia information.

More specifically, it will develop new formal models for knowledge representation with major focus being placed on multimedia ontological knowledge representation. Specifically, a multimodal data model will be constructed. Moreover, a whole work package will be devoted to the generation of an ontology infrastructure containing all the knowledge needed for the analysis in three main ontologies: the mathematical feature ontology (MFO) containing the knowledge about the mathematical features (low-level descriptors), the semantic feature ontology (SFO) containing the semantic information concerning the multimedia content (the actors, directors, etc) and the user profile ontology (UPO) covering the information about the user preferences and the usage history of a specific user. Finally, merging multiple types of digital data into an integral representation is one of the main objectives. Thus, a formal data model for integration of diverse multimedia content (meta-publication) will be designed.

Also, it will provide new tools to support automatic analysis, annotation, filtering and visualization of multimedia content to the extend that this is possible. Specifically, tools will be developed for semantic annotation of the existing multimedia by human experts, collaborative online and offline learning of document concepts and (semi-)automated annotation. Tools will also be developed for management and presentation of multimedia meta-publications.

The project maximizes cross-fertilization between several different areas, including knowledge technologies, database technology, multimedia processing and so on. M-ADVANTAGE brings together a diverse consortium that, as a whole, holds know how and acknowledged scientific expertise in a variety of areas. Within the project, research results, technical practices and tools provided by the partners will be integrated. In this process, among other things, the communication between the semantic approach, the statistical approach, and the Ontology definition approach will be studied and implemented. Statistical search will be used as a super set of the conventional methods in the sense that where these should fail it will always be possible with this methodology to grasp concepts embedded in images, text and videos together and deliver a complete content overview on somebody's search.

REFERENCES:

[1] J. Hunter, "Adding Multimedia to the Semantic Web - Building an MPEG-7 Ontology", International Semantic Web Working Symposium (SWWS), Stanford, July 30 - August 1, 2001

[2] RDF Schema Specification 1.0, W3C Candidate Recommendation 27 March 2000. http://www.w3.org/TR/rdf-schema/

[3] TV-Anytime Forum, http://www.tv-anytime.org/

[4] MPEG-21 Multimedia Framework, http://www.cselt.it/mpeg/public/mpeg-21_pdtr.zip

[5] NewsML http://www.newsml.org/

[6] ISO/IEC 15938-5 FCD Information Technology - Multimedia Content Description Interface - Part 5: Multimedia Description Schemes, March 2001, Singapore

[7] DAML+OIL Revised Language Specification, March 2001.
http://www.daml.org/2001/03/daml+oil-index

[8] J. Hunter, J. Drennan, S. Little "Realizing the Hydrogen Economy through Semantic Web Technologies", IEEE Intelligent Systems Journal - Special Issue on eScience, January 2004

[9] Little, S., and Hunter, J., "Rules-B-Example – a Novel Approach to Semantic Indexing and Querying of Images", In 3rd International Semantic Web Conference (ISWC2004), Hiroshima, Japan, November 2004.

[10] Tsinaraki, C., Polydoros, P., Christodoulakis, S., "Interoperability support for Ontology-based Video Retrieval Applications", In Proceedings of Third International Conference on Image and Video Retrieval (CIVR), Dublin, Ireland, July 21-23, pp 582-591, 2004.

[11] Baeza-Yates, R.A., Ribeiro-Neto, B.A. (1999) Modern Information Retrieval. ACM Press / Addison-Wesley.

[12] Zhao, R., W.I. Grosky (2002) Narrowing the Semantic Gap-Improved Text-Based Web Document Retrieval Using Visual Features. IEEE Transactions on Multimedia 4(2).

ICT OPPORTUNITIES AND CHALLENGES FOR REMOTE SERVICES

Jouni Pyötsiä
Metso Automation, Tulppatie 7 A, PO Box 310,
00811 Helsinki, Finland
jouni.pyotsia@metso.com

Abstract: Embedded intelligence together with upcoming ICT solutions give new possibilities for more automated service process operation over network during the life cycle of machines and systems. The purpose of this paper is to describe the existing solution based on Web Services and how it can be enhanced utilizing semantic web approach.

First, the paper describes automation and ICT challenges from business and technology point of view. Next, the paper presents a remote and networked service solution based on a business hub approach. The solution consists of service provider's Central Hub and several customers' Site Hubs, which have been integrated together over network. Finally, the benefits and future challenges of the new solution and semantic web utilization and approach are discussed.

Key words: Information technology, Intelligent machines, Embedded systems, Maintenance, Life cycles technology

1. INTRODUCTION

Rapid changes and discontinuities in the 21st century business environment will challenge companies. To ensure high flexibility, sustainable growth and profitability companies have to find new innovative business solutions. In many cases technology as such is not anymore sufficient to ensure competitive edge. New innovative business solutions call for strong integration of automation technology, information and communication technology (ICT) and business processes.

Especially the close integration requirements are true in new emerging remote and networked service solutions. Embedded intelligence in different machines and systems gives new possibilities for more automated business process operation over network during the machines and systems life cycle.

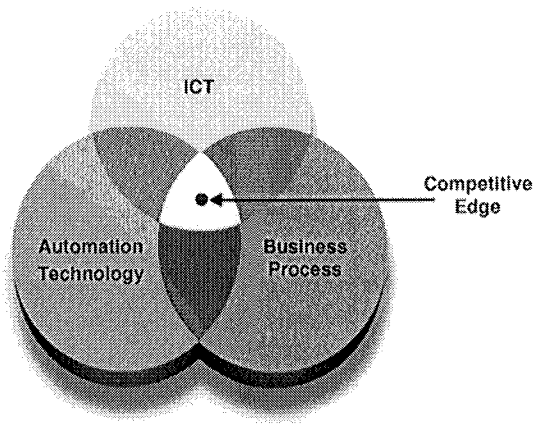


Figure 1. Ensuring competitive edge in future solutions means more close integration of different technologies and business processes.

The new emerging remote service solutions demand that products are transforming into life cycle services and these services are transforming into customers' service processes. Business messages coming from intelligent machines and systems drive these processes utilizing embedded intelligence and ICT solutions.

In the future the different collaborative resources like intelligent machines, systems and experts create huge amount of new data and information during the machines and systems life cycles. This information and message flow management and compression to on-line knowledge is also a demanding challenge.

On the other hand, optimization requirements demand more effective knowledge utilization and speed up network-based learning during the collaboration between different resources.

To overcome these demands we have to utilize Web Services, new upcoming semantic web based solutions and intelligent agent approach.

2. METSO'S STRATEGY AND ICT /1/

Metso Corporation is a global supplier of process industry machinery and systems as well as know-how and aftermarket services. The Corporation's core businesses are fiber and paper technology, rock and minerals processing and automation and control technology.

Metso's strategy is based on an in-depth knowledge of its customers' core processes, the close integration of automation and ICT, and a large installed base of machines and equipment.

Metso's goal is to transform into a long-term partner for customers. Metso will develop solutions and services to improve the efficiency, usability and quality of customers' production processes through their entire life cycles.

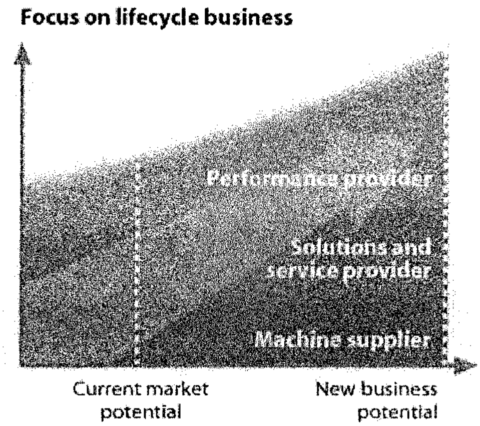


Figure 2. Metso's large installed base of machines and equipment creates a firm foundation for transformation into after market services.

Close co-operation between the customer makes it possible to optimize entire processes utilizing more embedded intelligence. Already in the design phase remote service capabilities can be embedded into machines and processes, which, in turn, form the basis for remote monitoring, process optimization and optimal maintenance and service solutions. Process optimization saves energy, raw materials and costs, minimizes emissions and environmental impacts and extends process life cycles.

ICT opens up the new possibility for remote and networked service business solutions and presents Metso the opportunity to develop new business models. These models must be based on the life cycle business thinking.

3. METSO'S ICT CHALLENGES /2/, /4/

The problems in today's networked business solutions and remote services start from security. Usually business partners have many point-to-point connection holes into their Intranet and have used even modem based connections for that. This usually means low security, difficult management and extra costs for the business partners.

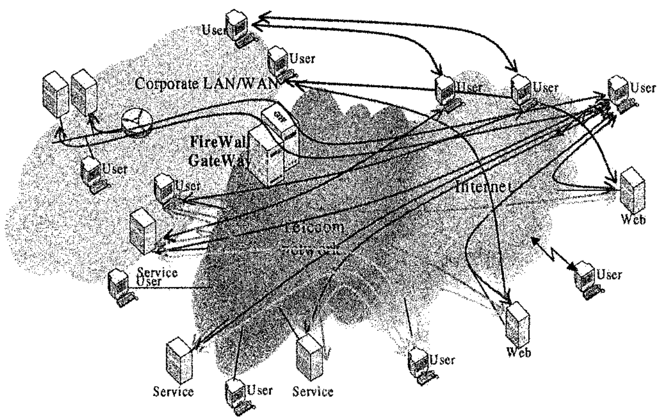


Figure 3. In many cases today connectivity is a real security risk for different partners in networked business environment.

Another problem is the point-to-point integration between different systems and applications. Business process automation, easy integration and business messages management are very difficult inside the company and almost impossible between business partners.

In today's business environment a key driver for a company's business strategy is its adaptation to a changing business environment. ICT must create a flexible and nimble business architecture based on security and cost efficiency to continuously resolve the highest advantage to the business.

From the technology and business point of view the following ICT challenges have to be met to make the required service business transformation a reality:

- Pervasive Communication:
  - Machine to machine (m2m)
  - Application to application (a2a)
  - System to system (s2s)
  - Collaboration (b2b)
- Network Security
  - High security and confidentiality

- Easy to build-up and maintain

- Industry Cluster Wide Standards:
  - Security
  - Messages
  - Protocols
  - Collaborative business processes
- On-line Customer Services:
  - Fast response
  - Better focus to real value
  - Network based learning
- Operational Excellence:
  - Strong cost reduction
  - Punctuality in communication
  - Fluent business process operation
  - Transparency

4. SOLUTION FOR REMOTE AND NETWORK SERVICES /4/

4.1 Business Hub

Business hub is a solution to provide secure VPN (Virtual Private Networks) connection between Metso and its customers' intelligent machines and systems. Standard and corporate wide security solution creates fast built, reliable and cost effective solution for customers' integration. Strong authentication, strong encryption and traceability of users and connections guarantee high security.

Enterprise Application Integration (EAI) platform is today's solution for collaboration and business logic (rules and process) modeling in the Business hub. Internal integration is always the starting point for more advantaged collaborated solutions. Figure 4 shows Metso's vision 2001 for EAI hub.

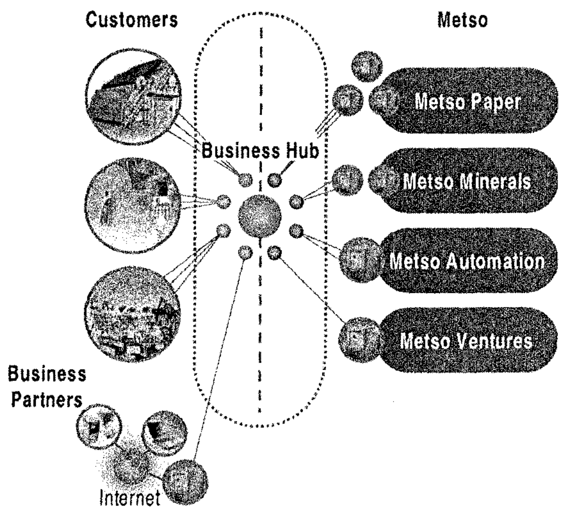


Figure 4. Business hub based on EAI technology is today's solution for collaboration between customers' intelligent machines and systems.

4.2 Web Services

Web Services are important building blocks for information exchange (SOAP), description (WSDL) and discovery (UDDI) between different resources (like applications, systems, machines and experts) in global network.

Web Services open up new possibilities to move valuable maintenance and process performance information from customer sites to Metso's Remote Service Centers and experts. The key lies in utilizing the intelligence embedded in installed base and automating the message flows between Metso and its customers' applications through the Internet. Web Services based messages and interfaces will allow machines and systems to communicate with each other independently and automatically over network.

Web Services together with EAI workflow and business process tools create powerful vehicle to automate the operation of remote and networked service solutions. Web Services Flow Language (WSFL) and Business Process Execution Language for Web Services (BPEL4WS) give new possibilities to use open XML based standard for business process and workflow descriptions.

Figure 5 shows workflow by which it is possible to automate different kinds of functions and operations concerning remote and networked services.

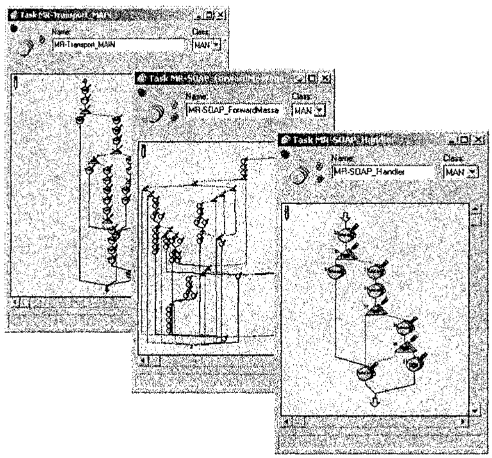


Figure 5. Ready-made "lego" modules by which it is easy and effective to build up service logic and more automated remote and networked service solutions.

4.3 Remote Service Solution Description

Metso's remote service solution consists of service provider's Central Hub and several customers' Site Hubs, which have been integrated together over network. SiteHub solution is based on an EAI Platform. In addition to basic EAI functionality the SiteHub includes the following features:

- Message management
- Application management
- User Management
- Message security
- System and application monitoring
- Enterprise wide site hub integration
- Partner network integration
- User interface support

The key issues in SiteHub solution are: open standards, information security, reliability, connectivity and manageability. These requirements are met by combining a traditional EAI platform with new features, which are specially designed upon industrial needs.

The factory floor connectivity is one of the main issues with SiteHub solution. While the isolated site operation is well supported, the main power of the information flow comes, when the applications are accessible in office / corporation levels. Open and standard interfaces provide easy access with applications.

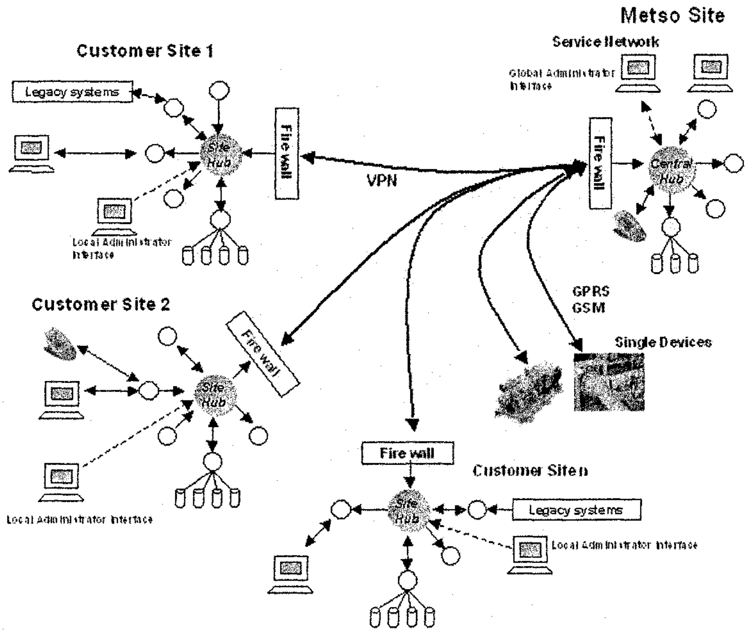


Figure 6. SiteHub network architecture.

4.4 SiteHub Architecture

The base building block of the SiteHub is Integration Platform. The operating logic of the hub is implemented as tasks. In addition to these tasks SiteHub consists of two external services: Management service and User Interface service. Management service is handling the management of users and applications connected to the system. User Interface service provides the applications a standard way to present their user interfaces.

Figure 7 presents the architecture and building blocks of the SiteHub solution.

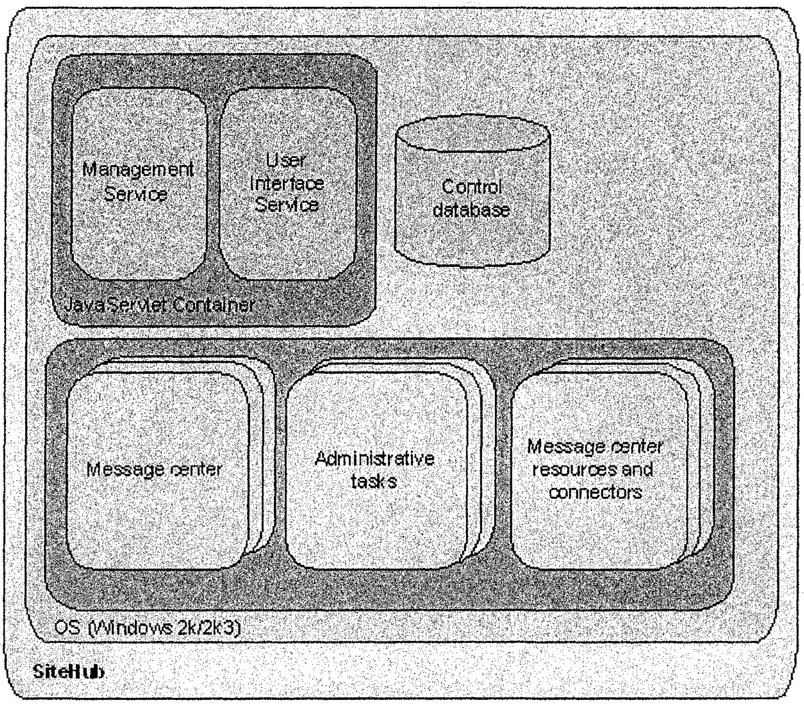


Figure 7. SiteHub architecture.

Message Center in Figure 7 is the main building block of the SiteHub. It takes care of receiving the messages, checking the validity of them and routing them to the correct receivers. Messaging in SiteHub is based on Web Services technology. The operation logic build on top of Integration Platform is working as a message center. It receives the incoming message and delivers the message to the receiving application or to another hub if the receiving application is connected to that.

Message Center supports two message delivery models: synchronous and asynchronous. In synchronous messaging the connection is opened between the sender and the receiver and the receiver has to answer to the message. In asynchronous messaging the Message Center answer to the sender right away it has received the message. In this case the Message Center guarantees that the message is delivered to the receiver. If the receiver cannot be connected the message is stored into the control database. Message Center tries the resending of the message until it can be delivered (or until a specified time has passed).

Connections from the SiteHub to partners and to Central Hub can be made securely with VPN connections.

5. FUTURE CHALLENGES /2/, /3/, /4/, /5/, /6/

Semantic web technology led by World Wide Web Consortium (W3C) gives totally new opportunities for building information and knowledge management solutions between different resources in networked business environment.

Integration of Web Services and new enabling semantic web technologies (like RDF, RDF(S), OWL and DAML-S) create comprehensive and more intelligent web services environment.

Resource Description Framework (RDF) provides interoperability and easier discovery between different resources that exchange information on the Web. RDF gives good basis for maintenance and performance information description and classification.

Web Ontology Language (OWL) describes the structure of knowledge and enables knowledge sharing and integration between resources. Ontology enhancement for Web Services is the most important remedy for present SiteHub solution.

DARPA Agent Markup Language for Services (DAML-S) describes the upper level ontology for properties and capabilities and it enables automatically discover, invoke, compose and monitor for web services.

With the help of agent technology it is possible to develop a simple and advanced performance evaluation and predictive maintenance concept for intelligent machines and devices. This concept is based on smart agents, a network of smart agents and self-learning capabilities.

The agent-based system determines the performance and health of machines and devices with the help of two indices: performance index and maintenance need index. The performance index is a key to evaluating the operation of machines and devices relating to the operational and control performance. The maintenance index is a key to predicting future needs for maintenance.

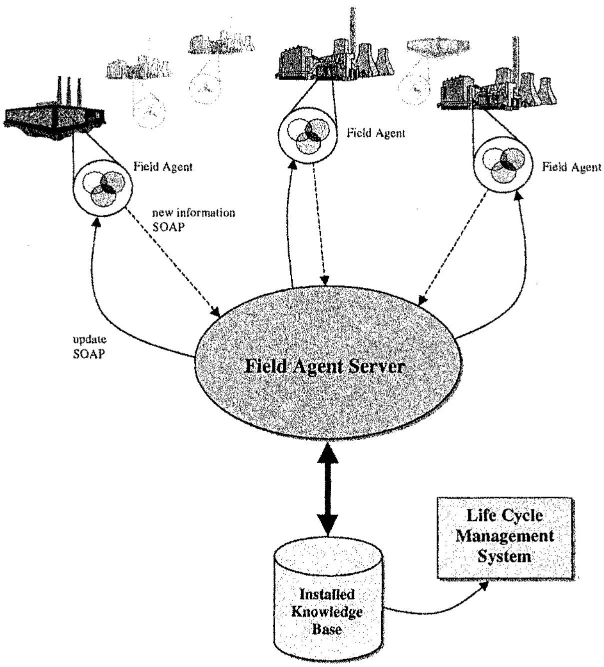


Figure 8. Field Agent network and server architecture.

Field Agent is a software component that automatically follows the performance and health of machines and devices. It is autonomous, it communicates with its environment and other Field Agents, and it is capable of learning new things and delivering new information to other Field Agents. The use of the Field Agent is invisible to the user. It delivers reports and alarms to the user by means of Web Services and new emerging semantic web technologies.

The emerging semantic web technologies give new possibilities also in implementation of Field Agent concept. Semantic Peer-to-Peer (P2P) architecture provides a direct interaction with other Field Agents over network for learning and service discovery.

In emerging remote and networked services the information and knowledge discovery and seamless sharing between different networked resources (applications, systems, intelligent machines and experts) are must. To overcome these challenges semantic web with field agent approach is an important step to more powerful solutions.

6. SUMMARY

It could be said that products and solutions are transforming into services and services are transforming into service business processes in networked business environment.

To make this vision true in remote and networked services we have to closely integrate lots of different kinds of technologies and business processes.

The existing Enterprise Application Integration (EAI) technologies, Web Services and upcoming semantic web technologies give new tools to make valuable integration and ontology description a reality.

New solutions based on the previously mentioned technologies will guarantee the increased yield, decreased total cost of ownership and improved safety through more powerful remote and networked service solutions. The key is that the right information and knowledge are at the right time in the right place in collaborated business environment.

Still a lot of work needs to be done especially in agent-based embedded intelligence and standardization. More capable intelligence is needed into the machines and systems. To create powerful proactive services we have to get more reliable reasoning and even network based learning to support decision making. On the other hand, standards convergence is a must in a more automated business process operation over network. Otherwise lots of adapters, conversions and transformations shall be made to applications, messages and processes between different business partners.

REFERENCES

/1/ Metso Corporation, Sustainability Report 2003, March 10, 2004, URL: http://www.metso.com

/2/ Pyötsiä J., Cederlöf H., Advanced Diagnostics Concept Using Intelligent Field Agents, ISA99, Philadelphia, USA.

/3/ Pyötsiä J., Cederlöf H., Remote Wireless Presence in Field Device Management, ISA EXPO 12-24 August 2000, New Orleans, USA.

/4/ Pyötsiä J., New Information and Communication Technology (ICT) Challenges in Remote and Networked Services, IFAC Symposium on Telematics Applications in Automation and Robotics, 21-23 June 2004, Helsinki University of Technology, Finland.

/5/ Terziyan V., Kononenko O., Semantic Web Enabled Web Services: State-of-Art and Industrial Challenges In: M. Jeckle and L.-J. Zhang (eds.), Web Services - ICWS-Europe 2003, Lecture Notes in Computer Science, Vol. 2853, Springer-Verlag, 2003, pp. 183-197.

/6/ Terziyan V., Semantic Web Services for Smart Devices Based on Mobile Agents, In: Forth International ICSC Symposium on Engineering Intelligent Systems (EIS-2004), Island of Madeira, Portugal, February 29 - March 2, 2004

MODELING ONTOLOGY VIEWS:
An Abstract View Model for Semantic Web

Rajugan, R.[1], Elizabeth Chang[2], Tharam S. Dillon[1], Ling Feng[3] and Carlo Wouters[4]
[1]eXel Lab, Faculty of IT, University of Technology, Sydney, Australia; [2]School of Information Systems, Curtin University of Technology, Australia; [3]Faculty of Computer Science, University of Twente, The Netherlands; [4]Department of Computer Science & Computer Engineering, La Trobe University, Australia

Abstract: The emergence of Semantic Web (SW) and the related technologies promise to make the web a meaningful experience. However, high level modeling, design and querying techniques proves to be a challenging task for organizations that are hoping to utilize the SW paradigm for their industrial applications. To address one such issue, in this paper, we propose an abstract view model with conceptual extensions for the SW. First we outline the view model, its properties and some modeling issues with the help of an industrial case study example. Then, we provide some discussions on constructing such views (at the conceptual level) using a set of operators. Later we provide a brief discussion on how such this view model can utilized in the MOVE [1] system, to design and construct *materialized* Ontology views to support Ontology extraction.

Key words: Semantic Web (SW), view models, Ontology views, Object-Oriented conceptual models (OOCM), conceptual views, Ontology extraction

1. INTRODUCTION

The emergence of Semantic Web (SW) and the related technologies promise to make the web a meaningful experience. Conversely, success of SW and its applications depends heavily on utilization and interoperability of well formulated Ontology bases in an automated, heterogeneous environment. This creates a need to investigate utilization of (materialized) Ontology views [2] in SW applications, such as; (a) Ontology extraction, (b) Ontology versioning, (c) sub-ontology bases, and (d) SW-wrappers for traditional data sources, in industrial settings. However, unlike traditional database systems, high level modeling, design and querying techniques still proves to be a challenging task for SW paradigm as, Ontology view definitions and querying have to be done at high-level abstraction [2, 3].

The databases systems (from relational to deductive systems) have matured enough to face growing challenges faced by the organizations (both commercial and governments) and their emerging (and aging) Enterprise Information Systems (EIS). They have well defined basic principles [4] on which they are built upon. Due to this, supporting data intensive technologies, such as transaction processing, business queries, data warehousing, data mining etc. have evolved to a level that can be considered as "matured". Many new and ongoing research directions in data intensive domains still follow the basic principles of databases [5], namely meta-data, schema and instance data. This, in our view is one of the major differences between the database and the SW principles, where meta-data schemas and instance data may overlap. Also, the data extraction process (e.g. queries), in direct contrast to user queries in database systems, is usually automated and involves meta-data extraction as part of the process.

On the other hand, Semantic Web directives are still at its infancy in areas such as data organization, meta-data models and query languages. As a result, in the present stage of SW developments, there are lots of contradictions than agreements in regards to basic concepts and definitions of the SW vocabularies (*see* section 2). Regardless of contradictions, many organizations, both academic and industry are working tirelessly in proposing new methodologies, models and are vigorously formulating standards to streamline the SW paradigm (some consider the present SW phase to be level 2 activities [1]).

On a positive note, there is an exponential growth in new research directions in SW applications. These applications range from SW-enabled traditional enterprise meta-data repositories to time-critical medical information and infectious decease classification databases. For such vast Ontology bases to be successful and to support autonomous computing in a distributed (and heterogeneous) environment, the preliminary design and engineering of such Ontology bases should follow a strict software engineering discipline [6]. Furthermore, supporting technologies for Ontology engineering such as data extraction, integration and organization have be matured to provide adequate modeling and design mechanism to build, implement and maintain successful Ontology bases. For such purpose, Object-Oriented (OO) paradigm seems to be ideal choice as it has been proven in many other complex applications and domains [7, 8].

During mid relational and early Object-Oriented (OO) revolution, during similar phase of the technological development and standardization (level 2), all agreed (both academia and industry) that the data models should be independent of the underlying language semantics and syntaxes and be able to provide needed abstraction and model portability [7, 9]. Today, this notion still holds true for SW paradigm.

To address such an issue, in this paper, we propose and *abstract* view model for modeling and designing views for SW paradigm (SW-view). Such abstract view model is defined using a high-level modeling OO language (such XSemantic net

[10, 11] or OMG's UML [12] or Ontology Web Language (OWL) [13], in contrast to Ontology specific data language) that is capable of modeling Ontologies.

The rest of this paper is organized as follows. In section 2, we briefly describe some of the terminologies used in the context of SW, followed by some of the early work done in view related domain in section 3. Section 4 describes our view model. Section 5 briefly outlines how our view model is utilized in the MOVE [1] system. In section 6 we provide some illustrative examples of our view model concepts that are based on a real-world industrial case study. Section 7 concludes the paper with some discussion on our future research directions.

2. DATABASES, ONTOLOGIES AND VIEWS

Databases and Ontologies serve to structure vast amount of information that is available at given point in time [14]. But in theory, there exists a clear distinction between databases and Ontologies, namely, the clear distinction between the schema and the instances. In databases (relational, OO, active, etc.), schemas are precisely defined in one level of abstraction (usually at the logical or schemata level) and instances are added, edited and/or validated in another layer. Usually views in database systems are defined as part of the external schema. Conversely, Ontology bases tend to have heterogeneous schemas at varying levels of abstraction (logical or instantiated schemas) and instances may co-exist among these schemas to convey information, concepts or relationship between two concepts to the users.

Another intriguing difference between database and Ontology base is that, database trend to follow a well-defined and established standard/(s), while Ontology standards, functionality and definitions trend to differ between implementations and models due to its infancy [2, 15]. For example, in OWL [16] one can create instances as part of the ontology but not in the DOGMA approach [6].

For the purpose of this paper, we need to make a distinction between the concept of abstract view definitions (addressed in this paper) for SW and the view definitions in SW languages such as Resource Description Framework (RDF) [17] and the Ontology Web Language (OWL, previously known as DAML+OIL) [16]. Though expressive, SW related technologies and languages suffer from visual modeling techniques, fixed models/schemas and evolving standards. In contrast, higher-level OO modeling language standards (with added semantics to capture Ontology domain specific constraints) are well-defined, practiced and transparent to any underlying model, language syntax and/or structure [18]. They also can provide well-defined models that can be transferred to the underlying implementation models with ease. Therefore for the purpose of this paper, an abstract view for SW is a view, where its definitions are captured at a higher level of abstraction (namely, conceptual), which turn can be transformed, mapped and/or materialized at any given level of abstraction (logical, instance etc.) in a SW specific language and/or model.

In addition, an abstract view model for SW should be able to deal with not just one but multiple data encoding language standards and schemas (such as XML, RDF, OWL etc.), as enterprise content may have not one, but multiple data coding standards and ontology bases. Another issue that deserves investigation is the modeling techniques of views for SW. Though expressive, SW related technologies suffer from proven visual modeling techniques [18]. This is because Object-Oriented (OO) modeling languages (such as UML) provide insufficient modeling constructs for utilizing semi-structured (such as XML, RDF, OWL) schema based data descriptions and constraints, while XML/RDF Schema lacks the ability to provide higher levels of abstraction (such as conceptual models) that are easily understood by humans. To address this issue, many researchers have proposed OMG's UML (and OCL) based solutions [2, 15, 18-21], with added extensions to model semi-structured data.

3. RELATED WORK

We can group the existing view models into four categories, namely; (a) classical (or relational) views [4, 22], (b) Object-Oriented (OO) view models [5, 23], (c) semi-structured (namely XML) view models [24-26] and (d) view models for SW. An extensive set of literature can be found in both academic and industry forums in relation to various view related issues such as (i) models, (ii) design, (iii) performance, (iv) automation and (v) turning and refinement, mainly supporting the 2-Es; data Extraction and Elaboration (with and some research directions towards 3-Es, i.e. 2-Es and data Extension). A comprehensive discussion on existing view models can be also found in [26]. Here, we focus only on view models for semi-structured data and SW.

Since the emergence of XML [27], the need for semi-structured data models to be independent of the fixed data models and data access, violates fundamental properties of the classical data models. Many researchers have attempted to solve semi-structured data issues by using graph based [28] and/or semi-structured data models [29, 30]. But, as in the case of relational and OO, the actual view definitions are only available at the lower levels of the implementation and not at the conceptual and/or logical level [26, 31].

One of the early discussion on XML views was by Serge Abiteboul [24] and later more formally by Sophie Cluet et al. [32]. They proposed a declarative notion of XML views. Abiteboul et al. pointed out that, a view for XML, unlike classical views, should do more than just providing different presentation of underlying data [24]. This, he argues, arises mainly due to the nature (semi-structured) and the usage (primarily as common data model for heterogeneous data on the web) of XML. He also argues that, an XML view specification should rely on a data model (like ODMG [33] model) and a query language. In the paper [32], they discuss in detail on how abstract paths/DTDs are mapped to concrete paths/DTDs. These concepts, which are implemented in the Xyleme project [34, 35], provide one of the most comprehensive mechanisms to construct an XML view to-date. The Xyleme project uses an extension of ODMG Object Query Language (OQL) to implement such an XML view. But, in relation to conceptual modeling, these view concepts provide no support. The view model is derived from the instantiated XML documents (instant level) and is associated with DTD in comparison to flexible XML Schema. Also, the Xyleme view concept is mainly focused on web based XML data.

Another XML view model; the MIX (Mediation of Information using XML) [36] view system, is a by-product of developing web scale mediator systems. The MIX system is based on mediator architecture supporting to provide the user with an integrated view of the underlying heterogeneous information/data sources. The MIX system employs XML as the data exchange and integration medium between mediator components and the XML DTD to provide structural descriptions of the data. Though MIX system provides support for XML views, it is not an XML view by nature. It is a by-product to support data mediation for web-based information systems. Though powerful, the drawback includes no standalone framework to support XML views and non-standard language/(s) used to query/manipulate data.

Another view model for XML, which is based on Object-Relationship-Attribute model for Semi-Structured data (ORA-SS) was proposed by authors in [25]. It is an intuitive data model for XML based on Entity-Relationship (ER) model and the static OO model. An object in ORA-SS is similar to that of an entity in ER (similar to that of an XML element), while a relationship is similar to that of a relationship between two entities in ER. Attributes of ORA-SS describe the objects and relationships. This is one of the first view model that supports some of abstraction above the data language level.

In the work [26, 31], we proposed a layered view model for XML with three levels of abstraction, namely; conceptual, logical and instance levels. In the view model, the view definitions are captured at the conceptual level using a set of conceptual operators [37]. The conceptual view definitions are transformed to logical/schema view definitions (using XML schema definition language) and to document/instance view query expressions (e.g. such as XQuery and or SQL 2003). An added advantage of such view model include; (a) capture conceptual semantics that are easily understood by both human and machines (in contrast to machine-friendly query expressions), (b) view definition are independent of any query language syntax, (c) provide view validation using XML (view) schema and (d) expressive view semantics that support extraction, elaboration or both.

In related work in Semantic Web (SW) [38] paradigm, some view models have been proposed in [3, 39], where the authors present a RDF views with support for RDF [17] schema (using a RDF schema supported query language called RQL). This is one of the early works focused purely on RDF/SW paradigm and has sufficient support for logical modeling of RDF views. The extension of this work (and other related projects) can be found at [40]. RDF is an object-attribute-value triple, where it implies object has an attribute with a value [41]. It only makes intentional semantics and not data modeling semantics. Therefore, unlike generic view models, views for such RDF (both logical and concrete) have no tangible scope outside its domain. In related area of research, the authors of the work proposed a logical view formalism for ontology [1, 15, 42] with limited support for conceptual extensions, where materialized ontology views are derived from conceptual/abstract view extensions.

Another area that is currently under development is the view formalism for SW Meta languages such as OWL. In some SW communities, OWL is considered to be a conceptual modeling language for modeling Ontologies, while some others consider it to be a crossover language with rich conceptual semantics and RDF like schema structures [1]. It is outside the scope of this paper to provide argument for or against OWL being a conceptual modeling language. Here, we only highlight one of view formalism that is under development for OWL, namely views for OWL in the "User Oriented Hybrid Ontology Development Environments" [43] project.

4. OUR ABSTRACT VIEW MODEL FOR SEMANTIC WEB

In this paper, we present an abstract view model with conceptual extensions for the SW (SW-view). Initially such view model was proposed for XML by us in [26, 31], with clear distinction between three levels of abstraction namely; (a) conceptual, (b) logical (or schematic) and (c) document (or instance). Here it is adopted for the SW paradigm.

In work with XML, we provided clear distinction between conceptual, logical and document levels views, as in the case of data engineering, there exists a need to clearly distinguish these levels of abstractions. But in the case of SW domain, though there exists a clear distinction between conceptual and logical models/schemas, the line between the logical (or schema) level and document (or instance) level trends to overlap due to the nature of the data sources (namely Ontology bases), where concepts, relationships and values may present mixed sorts, such as schemas and values [14]. Therefore, in the SW-view model, we provide a clear distinction between conceptual and logical views, but depending on the application, we allow an overlap between logical views and document views. This is one of the main differences between the XML view model and the SW-views.

To our knowledge, other than our work, there exist no research directions that explore the conceptual and logical view model for the Semantic Web (SW) paradigm. This notation of SW-view model has explicit constraints and an extended set of expressive conceptual operators to support Ontology extraction in the MOVE [1, 2, 15] system.

4.1 Conceptual Views

The conceptual views are views that are defined at the conceptual level with conceptual level semantics using higher-level modeling languages such as UML. To understand the SW-view and its application in constructing ontology views, it is imperative to understand its concept and its properties. First, an informal definition of the view concept is given followed by a formal definition that serves the purpose of highlighting the view model properties and the modeling issues associated with such a high-level construct.

Definition 1: A **conceptual view** is the one which is defined at the conceptual level with higher level of abstraction and semantics.

One such abstract view model will; (i) provide data abstraction to view data set similar to a class (in OO) does to real-world objects, (ii) enable the software designers (not the programmers) to visualise, construct and validate constructed data sets (views) that are normally left to implementers, (iii) utilise as a tool to communicate better with the domain users and to improve domain user feedback (as users usually used to visualise data as a constructed data sets (views) than a stored/modelled data) and (iv) be utilised by system designers to add additional data semantics at a higher level of abstractions to data intensive domains (such as SW or XML domains), where both data and data semantics are important.

4.2 Conceptual View Properties

To utilize the SW-view model in applications, it is imperative that, one must first understand some of its unique properties and characteristics. In this section, we first provide some of the SW-view formal semantics followed by the derivation of the conceptual view definition. It should be note here that, though there can be more elaborated definitions are possible depending on the application domain, here we provide a simplified generic SW-view definition that can be easily applied to ontology extraction. Following the conceptual view definition are the sections that address some of the unique characteristics of the SW-views, conceptual operators, some modeling issues and the descriptive constraint model.

Conceptual Objects (CO): CO refers to model elements (objects, their properties, constraints and relationships) and their semantic inter-relationships (such as composition, ordering, association, sequence, all etc) captured at the conceptual level, using a well-defined modeling language such as UML, or XSemantic nets [10, 11], OWL or E-ERD [4] etc. A CO can be either of type simple content ($s_{content}$) or complex content ($c_{content}$) depending on its internal structure [10, 41, 44]. For example, CO that uses primitive types (such as integer, character etc) as their internal structure corresponds to $s_{content}$ and CO that uses composite objects represent their internal structure corresponds to $c_{content}$.

Conceptual Schema (CS): We refer conceptual schema as the meta-model (or language) that allow us to define, model and constrain COs. For example, the conceptual schema for a valid UML model is the MOF. Also, the UML meta-model provides the namespace of such schemas.

Like XML/RDF Schema, where the instance will be an XML/RDF document, here, an instance of the conceptual schema will be a *well-defined, valid* conceptual model (in this case in UML) or other conceptual schemas (i.e. such as MOF), which can be either visual (such as UML class diagrams) or textual (in the case of UML/XMI models).

Logical/Schema Objects (LO): When CO are transformed or mapped into the logical/schema level (such rules and mapping formalism described in works such as [10, 21, 41, 45, 46]), the resulting objects are called LO. These objects are represented in textual (such as a schema language, OWL) or other formal notations that support schema objects (such as graph).

Postulate 1: A ***context*** (ς) is an item (or collection of items) or a concept that is of interest for the organization as a whole. It is more than a measure [47, 48] and is a meaningful collection of model elements (classes, attributes, constraints and relationships) at the conceptual level, which can satisfy one or more organizational perspective/(s) in a given domain. Simply said, it is a collection of concepts, attributes and relationships that are of interest in construction of other ontology/(ies).

Postulate 2: A ***perspective*** (∂) is a viewpoint of an item (or a collection of items) that makes sense to one or more stakeholders of the organization or an organizational unit, at given point in time. That is, one viewpoint of a context at a given point in time.

Definition 2: A ***conceptual view*** ($\mathcal{V}_{co}$) [31] is a view, defined over a collection of *valid* model elements, at the conceptual level. That is, it is a perspective for a given context at a given point in time.

Let $\mathcal{X}$ be a collection of COs. Let $\Re$ be the rule set, constraints and syntaxes that makes $\mathcal{X}$ a ***valid*** collection of CO (according to a meta-modeling language such as MOF or UML or XSemantic nets). Therefore it can be shown that, a *valid conceptual collection set* $\mathcal{X}$ is a function of $\Re$, shown as;

$$\mathcal{X} = \Re(\mathcal{X}) \tag{1}$$

We can show that, a *valid* **conceptual view** [14] ($\mathcal{V}_{co}$) of the valid CO set **collection** $\mathcal{X}$ is defined as the **perspective** ∂ constructed over a **context** ς by the conceptual **construct** λ. The resulting conceptual view belongs to the **domain** $\mathcal{D}(\mathcal{V}_{co})$, (where $\mathcal{D}(\mathcal{V}_{co}) = \mathcal{D}_{co}(\varsigma)$) with **schema** $\mathcal{S}_{co}(\mathcal{V}_{co})$, (where $\mathcal{S}_{co}(\mathcal{V}_{co}) = \mathcal{S}_{co}(\partial)$). The conceptual view is said to be valid if it is a valid instance of the view schema $\boldsymbol{\mathcal{S}}_{co}(\partial)$. Therefore **conceptual view** $\mathcal{V}_{co}$;

$$\mathcal{V}_{co} = (\varsigma, \partial, \lambda, \mathcal{X}) \tag{2}$$

where; (a) the *view name* of $\mathcal{V}_{co}$ is provided by the perspective ∂, (b) the *domain* and the *namespace* for $\mathcal{V}_{co}$ is provided by the context ς in the *valid* CO collection set of $\mathcal{X}$, (c) the view construction is provide by the conceptual construct λ; i.e. *conceptual operators* that construct the view over a given context, (d) the *valid* collection set $\mathcal{X}$ set provides the data for the view $\mathcal{V}_{co}$ instantiation, (e) the *view schema* $\mathcal{S}_{co}(\mathcal{V}_{co})$ that constrains and validates the view instances of the view $\mathcal{V}_{co}$ and (f) the domain $\mathcal{D}(\mathcal{V}_{co})$ provides the domain for the view $\mathcal{V}_{co}$. Another equivalent form of this definitions can be found in our work in [26].

As we stated earlier, unlike XML-view model, the distinction between conceptual and logical levels are clearly state for SW-views, but not between logical and document views. A detailed discussion of this work can be found in [14].

4.3 Conceptual View Operators

The conceptual constructor is a collection of binary and unary operators, that operates on CO (at the conceptual level) to produce result that is again a valid CO collection. The set of binary and unary operators provided here is a complete or basic set; i.e. other operators, such as division operator [4] and compression (*see* section 6) can be derived from these basic set of operators.

4.3.1 Conceptual Binary Operators

The conceptual set operators are binary operators that take in two operands produces a result set. The following algebraic operators are defined for manipulation of CO collection sets. A CO collection set can be represented in UML, XSemantic nets or other high-level modeling languages.

Let x, y be two valid CO collection sets (operands) that belongs to domains $\mathcal{D}_{co}(x) = dom(x)$ and $\mathcal{D}_{co}(y) = dom(y)$ respectively.

1. *Union Operator*: A Union operator $\bigcup_{(x,y)}$ of operands x, y produces a CO collection set $\mathcal{R}$, such that $\mathcal{R}$ is again a valid CO collection that includes all COs that are either in x or in y or in both x and y with no duplicates. This can be shown as in (3) below, where $dom(\mathcal{R}) = \mathcal{D}_{co}(x) \cup \mathcal{D}_{co}(y)$.

$$\bigcup_{(x,y)} = \mathcal{R} = x \cup y = x \cup x' = y \cup y' \tag{3}$$

2. *Intersection Operator*: An *Intersection* operator $\bigcap_{(x,y)}$ of operands x, y produces a CO collection set $\mathcal{R}$, such that $\mathcal{R}$ is again a valid CO collection that includes all COs that are in both x and y.

$$\bigcap_{(x,y)} = \mathcal{R} = x \cap y \tag{4}$$

where $dom(\mathcal{R}) = \mathcal{D}_{co}(x) \cap \mathcal{D}_{co}(y)$. *Note*: Since both Union and Intersection operators are commutative and associative, they can be applied to n-ary operands.

3. *Difference Operator*: A *Difference* operator $\overline{D_{(x,y)}}$ of operands x, y produces a CO collection set $\mathcal{R}$, such that $\mathcal{R}$ is again a valid CO collection that includes all COs that are in x but not in y.

$$\overline{D_{(x,y)}} = \mathcal{R} = x - y \tag{5}$$

where $dom(\mathcal{R}) = \mathcal{D}_{co}(x)$. Also note; the difference operator is NOT commutative.

4. *Cartesian product Operator*: A *Cartesian product* operator $\times_{(x,y)}$ of operands x, y produces a CO collection set $\mathcal{R}$, such that $\mathcal{R}$ is again a valid CO collection that includes all COs of x and y, combined in combinatorial fashion.

$$\times_{(x,y)} = \mathcal{R} = x \times y \tag{6}$$

where $dom(\mathcal{R}) = \mathcal{D}_{co}(x) \times \mathcal{D}_{co}(y)$

5. *Join Operator*: A *Join* operator can be shown in its general form as;

$$\triangleright\triangleleft_{(x,y)} = \mathcal{R} = x \triangleright\triangleleft_{[j_{condition}]}\, y$$

where, optional join-condition provides meaningful merger of COs. A join-condition $j_{condition}$ be of the form; (1) simple-condition: where the join-condition $j_{condition}$ is specified using CO simple content $S_{content}$ types, (2) complex-condition: where the join-condition $j_{condition}$ is specified using CO complex content $C_{content}$ types and (3) pattern-condition: where the join-condition $j_{condition}$ is specified using a combination of one or more CO simple and complex content types in a hierarchy with additional constraints, such as ordering etc.

(i) Natural Join

A natural join operator $\triangleright\triangleleft_{(x,y)}$ of operands x, y is a join operator with no join-condition specified, produces a CO collection set $\mathcal{R}$, such that $\mathcal{R}$ it is equivalent to a Cartesian product operator. This can be shown as;

$$\triangleright\triangleleft_{(x,y)} = R = x \triangleright\triangleleft y = \times_{(x,y)} \tag{7}$$

(ii) Conditional Join

A join operator $\triangleright\triangleleft_{(x,y)}$ of operands x, y with explicit join-condition $j_{condition}$ specified produces a CO collection set $\mathcal{R}$, such that $\mathcal{R}$ will have only the combination of CO collection set that satisfies the join condition. The join-condition $j_{condition}$ can only be of type; (1) simple-condition and (2) complex-condition. This join is comparable to the relational operator θ join. This can be shown as;

$$\triangleright\triangleleft_{(x,y)} = \mathcal{R} = x \triangleright\triangleleft_{(j_{condition1}[AND.....])}\, y \tag{8}$$

(iii) Pattern Join

A join by pattern $\triangleright\triangleleft_{(x,y)}$ is a join by condition operator where the join-condition $j_{condition}$ is of type pattern-condition.

4.3.2 Conceptual Unary Operators

We propose four unary conceptual operators to construct conceptual views without loss of CO semantic that are represented in the model. The four conceptual operators are projection, selection, rename, and restruct(ure).

1. *PROJECT Operator*: Given a valid CO collection set x, and a set of CO (either $s_{content}$ or $c_{content}$ or combination of both $s_{content}$ and $c_{content}$), the project operator $\Pi_{(x)}$ will produce a CO collection set $\mathcal{R}$ where it has only the specified CO set with; (a) persevered node hierarchy, (b) preserved node order and (c) preserved semantic relationships (if any). If need to , the projected CO set (in the case of hierarchical CO/(s) can be specified using the W3C XPath [49] standard.

$$\Pi_{(x)} = \mathcal{R} = \Pi_{(CO_1, CO_2, \ldots\ldots)}(x) \tag{9}$$

where the domain of $\mathcal{R}$ is $dom(\mathcal{R}) = \bigcup_{k=1}^{m} dom(CO_k)$

2. *SELECT Operator*: Given a valid CO collection set x, the select operator $\sigma_{(x)}$ will produce a CO collection set $\mathcal{R}$, where it contains one or more matching CO (or collection) that satisfy the select-condition $s_{condition}$. In addition, the select-conditions can be combined using the AND, OR, NOT logical operators.

$$\sigma_{(x)} = \mathcal{R} = \sigma_{s_{condition}}(x) \tag{10}$$

Again, here, the select-condition $S_{condition}$ be of the form; (1) simple-condition: where the select-condition $S_{condition}$ is specified using CO simple content $S_{content}$ types and the select operator is called *value-based*, (2) complex-condition: where the select-condition $S_{condition}$ is specified using CO complex content $C_{content}$ types and the select operator is called *structure-based* and (3) pattern-condition: where the select-condition $S_{condition}$ is specified using a combination of one or more CO simple and complex content types in a hierarchy with additional constraints, such as ordering etc, where the select operator is called *structure-based.*

3. *RENAME Operator*: Given a valid CO collection set x, and a CO src (with old and new labels $(l^{old}, l^{new}) \in L_{able}$), the rename operator $\rho_{(x)}$ will return x where the label of src is changed. A RENAME operation *cannot*; (a) alter src specific data types and (b) alter src specific contents, values or constraints.

$$\rho_{(x)} = \mathcal{R} = \rho_{src\,(l^{old}, l^{new})}(x) \tag{11}$$

4. *RESTRUCT(ure)Operator*: Given a CO collection set x, and a CO, src (with a pair of positions, old and new (pos_1, pos_2)), where the positions can be either absolute or relative (in a CO hierarchy), the restructure operator $\delta_{(x)}$ will return $\mathcal{R}$, where the position of src (src can be either $s_{content}$ or $c_{content}$) is changed from pos_1 to pos_2.

$$\delta_{(x)} = \mathcal{R} = \delta_{src\,(pos_1, pos_2)}(x) \tag{12}$$

But a restructure operation does not allow; (a) deletion of CO/(s) in the hierarchy, (b) alter CO structural relationships, constraints, names or cardinality and (c) alter CO data type or values.

Note: The operators presented above are referred to as extended or non-restive *basic set*, as many secondary (e.g. DIVISION operator) and restrictive operators (*see* section 5) can be derived by combining one or more of these binary and unary operators.

4.4 Modeling Conceptual Views for SW

In this paper, to model conceptual views, we propose OMG's UML (for modelling Ontologies). The only purpose we use this notation is to demonstrate our concepts and applications, and not to emphasis or promote this as the only modeling notation for conceptual views.

UML [12] has established itself as the *defacto* modelling language of choice in OO conceptual modelling paradigm. UML provides a well defined rich collection of tools to visually model a given domain into needed level of abstraction. It can be said that, UML helps to provide a well-defined blue print for a software system that is easily understood both by users and developers alike. UML also provides extensibility to the modelling language in the form of *stereotypes* which we utilise in defining our *conceptual views*. In the case of Ontology engineering, UML provide classes (similar to concepts in ontology), attributes and relationships that are used in defining Ontology models [2] in this paper.

Another reason we adopt UML is that, its models are portable, i.e. many schemata transformation rules and mapping techniques exists for transforming UML models to [20, 21, 41]; (a) XML Schema, (2) Ontology Web Language (OWL), (c) RDF and (d) XMI. Therefore, for the purpose of this paper, UML is visual modelling language of choice for OO conceptual modelling and supports abstraction from classical data models to ontology bases.

4.5 Conceptual View Constraints

In data modeling, specifications often involve constraints. In the case of views, it is usually specified by the data languages (and mostly excluding constraints associated with data semantics) in which they are defined in. For example, in relational model, views are defined using SQL and a limited set of constraints can be defined using SQL[4, 22], namely, (i) presentation specific (such as display headings, column width, pattern order etc), (ii) range and string patterns for aggregate fields, (iii) input formats for updatable views, and (iv) other DBMS specific (such view materialization, table block, size, caching options etc).

In Object-Relational and OO models, views had similar constraints but they are more extensive and explicit due to the data model (yet data language dependent). The OO views are constructed and specified using DBMS specific (such as OQL[33]) and/or external languages (such as C++, Java or O_2C[23]). It is a similar situation in views for semi-structured data paradigm, where rich set of view constrains are defined using languages such as OQL based LOREL [50, 51]. Today, in the case of Ontology engineering (and in Ontology views), this is still holds true, where constraints are specified using programming modules than at the schemata and/or logical level. In doing so, the constraints are implicit and mostly accessible only at runtime of the system and not at the modeling and/or design time.

But the work by authors of [25] provides some form of higher-level view constraints (under ORA-SS model) for XML views, while the work in [3] provides some form of logical level view constraints to be defined in views for in SW/RDF paradigm. As our conceptual view mechanism is defined at a higher-level of abstraction, we can provide an explicit view constraint specification model, as most high-level OO languages (such as UML, XSemantic nets, E-ER) provide some form of constraint specification.

Here, for our view model, we look into using UML/OCL [52] as our view constraint specification language. Also, our work should not be confused with work such as [53], where authors use OCL to "model" (not to specify) relational views (in contrast to ontology views), which utilizes OCL from a data modeling point of view. In UML, the Object Constraint Language (OCL), which is now a part of the UML 2.0 standard, can support unambiguous constraints specifications for UML models including specification of ontology model elements . In our conceptual view model, we incorporate OCL (in addition to built-in UML constraint features) as our view constraint specification language to explicitly state view constraints. It should be noted that, we do not use OCL to define views, rather state additional constraints using OCL. OCL supports defining *derived* classes [52, 54], which is close to a view concept [53]. Some examples of UML/OCL constraints for conceptual views are given in section 6, below.

5. CONCEPTUAL VIEWS AND THE MOVE [1] SYSTEM

In the sections 4 above, we have shown how conceptual views can be constructed in a given industrial settings. Here, we briefly discuss how such views can be applied in Ontology extraction in the Materialized Ontology View Extractor (MOVE) system [1]. The MOVE system was initially proposed by Wouters et al. [1, 2, 15], for the construction *optimized* materialised Ontology views, with emphasis on automation and quality of the views generated. The MOVE view process includes model and design of conceptual views with the utilization of restricted conceptual operators in deriving materialized Ontology views. Some of the restricted view operators include [2, 14]; (a) synonymous rename (2) selection and (3) compression.

Definition 3: [14] (Informal) A Strict Semantic Web View (or Ontology View) is a materialized SW-view that is derived from an Ontology (called the base ontology). The derivation can consist of any (combination) of the following operations; synonymous rename, selection and compression.

6. AN ILLUSTRATIVE INDUSTRIAL CASE STUDY

The e-Sol Inc. aims to provide logistics, warehouse, and cold storage space for its global customers and collaborative partners. The e-Sol solution includes a standalone and distributed Warehouse Management System (WMS/e-WMS), and a Logistics Management System (LMS/e-LMS) on an integrated e-Business framework called e-Hub [55] for all inter-connected services for customers, business customers, collaborative partner companies, and LWC staff (for e-commerce B2B and B2C). Some real-world applications of such company, its operations and IT infrastructure can be found in [55-57]. Here, use this system as the base to model and integrate (using Ontology views) Ontology bases and various sub-ontology vocabularies used at various customer and collaborative partner locations.

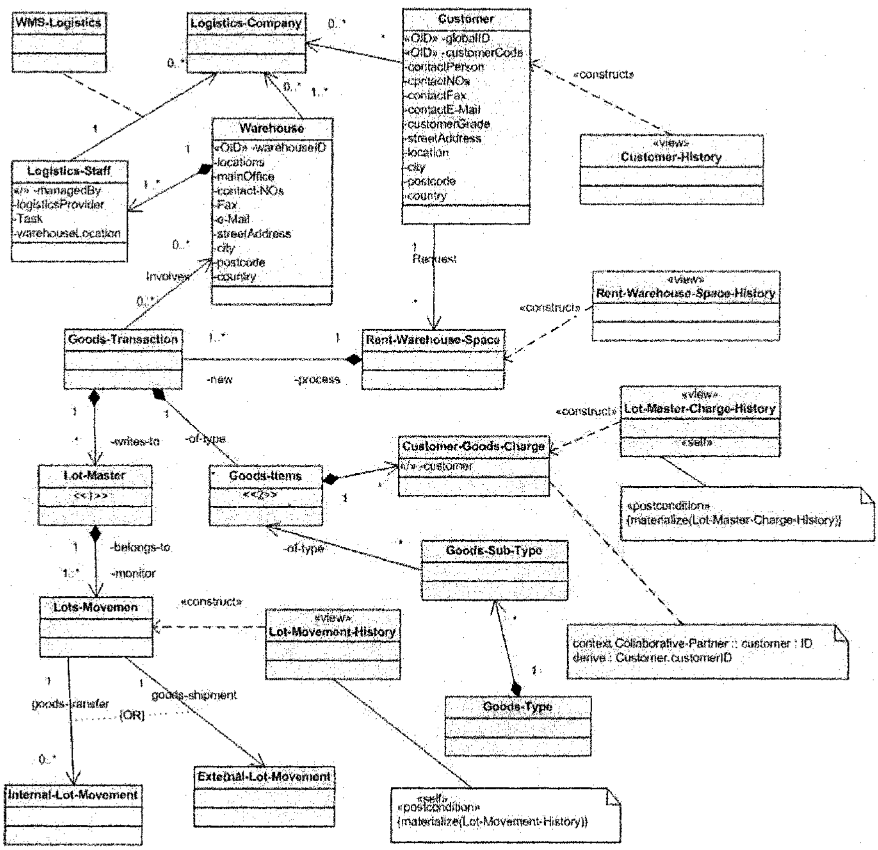


Figure 1. e-Sol example, Core Data Store Model (UML/OCL)

In WMS (Fig. 1-3), customers book/reserve warehouse and cold storage space for their goods. They send in a request to warehouse staff via fax, email, or phone, and depending on warehouse capacity and customers' grade (individual, company or collaborative partner), they get a booking confirmation and a price quote. In addition, customers can also request additional services such as logistics, packing, packaging etc. When the goods physically arrive at the warehouse, they are stamped, sorted, assigned lots numbers and entered into the warehouse database (in Lots-Master). From that day onwards, customers get regular invoices for payments. In addition, customers can ask the warehouse to handle partial sales of their goods to other warehouse customers (updates Lots-Movement and Goods-Transfer), sales to overseas (handled by LMS) or take out the goods in full or in partial (Lots-Movement). Also customer can check, monitor their lots, buy/sell lots and pay orders via an e-Commerce system called e-WMS. In LMS, customers use/request logistics services (warehouse or third-party logistics providers) provided by the warehouse chains. This service can be regional or global including multi-national shipping companies. Like e-WMS, e-LMS provide customers and warehouses an e-Commerce based system to do business. In e-Hub, all warehouse services are integrated to provide one-stop warehouse services (warehouse, logistics, auction, goods tracking, payment etc) to customers, third-party collaborators and potential customers.

In e-Sol, due to the business process, data semantics have to be in different formats (Ontology bases and vocabularies) to support multiple systems, customers, warehouses and logistics providers. Also, data have to be duplicated at various points in time, in multiple databases, to support collaborative business needs. In addition, since new customers/providers to join the system (or leave), the data formats has to be dynamic and should be efficiently duplicated without loss of semantics. This presents an opportunity to investigate how to integrate and utilize various customers' and collaborative partners' Ontology bases for mutual benefit and SW applications. The following example highlights some example of conceptual views developed for e-Sol. ***Note:*** It should be note that, the examples and the figures given for the e-Sol are demonstration purpose only and do not provide the complete Ontology base model of the system.

Example 1: Context (in Fig. 1-2), "`staff`", "`order`", and "`customer`" can be some of the context examples in the e-Sol system.

Example 2: Conceptual views (Fig. 1), "`Customer-History`", "`Lot-Master-Charge-History`" and "`Rent-Warehouse-Space-History`" are perspectives / views in the context of "`Warehouse-History`" of the e-Sol system.

Example 3: Conceptual view (Fig. 2), "`Collaborative-Partner`" is a perspectives / view in the context of "`Customer`" in e-Sol.

Example 4: Conceptual views, for example, "`processed-order`" and "`overdue-order`" are two contrasting views in the context of "`order`" of the e-Sol system.

Example 5: In Fig. 2, "`Warehouse-Manager`" is a valid XML conceptual view, named in the context of "`Staff`". It is constructed using the conceptual SELECT operator, which can be shown as;

$$\sigma_{\texttt{warehouse-Staff.Role="manager"}}(\texttt{Core-Users}).$$

Example 6: Similarly (Fig. 2), "`Site-Manager`" is a perspective/ view in the given context of "`Warehouse-Manager`".

Example 7: Another valid conceptual view "`Lot-Master-Charge-History`" in the given context of "`Warehouse-History`". Here, at the conceptual level, it is stated as a *materialized* conceptual view, implying that it is a persistence view during the life time of the system. This characteristic is also stated in the OCL statement (Fig.1).

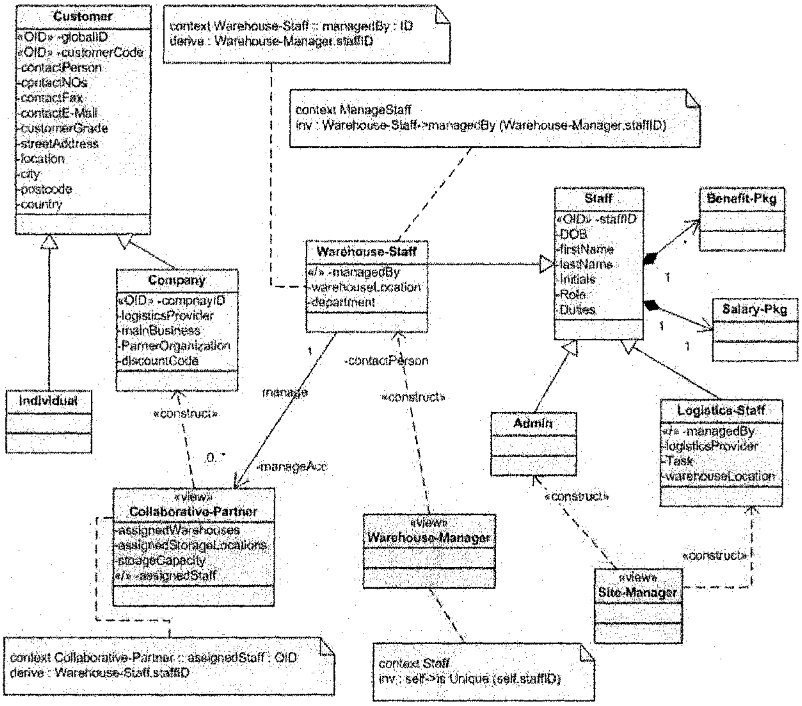


Figure 2. e-Sol, Business User Model (UML/OCL)

Example 8: In the case of conceptual view "`Warehouse-Manager`" (Fig. 2), we indicate the unique `staffID` by the following OCL expression;

```
context Staff
inv : self->isUnique(self.staffID)
```

Example 9: In the case of conceptual view "`Income`" (Fig. 3), the following OCL statements hold true;

```
context Income :: Staff : ID
derive : Staff.staffID

context Income :: benefits : Real
derive : Benefit-Pkg.totalBenefits

context Income :: baseSalary : Real
derive : Salary-Pkg.baseSalary
```

```
context Income :: totalSalary : Real
derive : totalSalary =
    (self.baseSalary - self.tax)
     + benefits
     - self.totalDeductions
```

Example 10: In the case of conceptual views "`Warehouse-Manager`" and "`Warehouse-Staff`", in the context of "`Staff`" (Fig. 2), we indicate the adhesion relationship between them using the following OCL statements given below.

```
context Warehouse-Staff :: managedBy : ID
derive: Warehouse-Manager.staffID

context Warehouse-Manager
inv: self.responsibleFor := Set(Warehouse-Staff.staffID)

context ManageStaff
inv : Warehouse-Staff->managedBy (Warehouse-Manager.staffID)
```

Example 11: In the case of conceptual views "`Lot-Movement`" (Fig. 1), the exclusive disjunction between `Internal-Lot-Movement` (stored goods change owners) and `External-Lot-Movement` (goods shipped outside the warehouse) can be show via the OCL statement "`OR`" between the relationships as shown in Fig. 1.

Example 12: If a new domain requirement exists to add new conceptual view "`Management-Memo`" send to all "`Warehouse-Manager`", we can do that using Cartesian Product conceptual operator, where x = `Warehouse-Manager` and y = `Management-Memo`;

$$\times_{(x,y)} = \mathcal{R} = x \times y$$

Example 13: In the case of conceptual view "`Income`" (Fig. 3), the conceptual construct is a conceptual JOIN operator with join conditions, where x = `Staff`, y = `Salary-Pkg` and z = `Benefit-Pkg`:

$$\mathcal{R} = (x \rightarrow_{(x.staffID=y.staffID)} y) \ AND \ (x \rightarrow_{(x.staffID=z.staffID)} z)$$

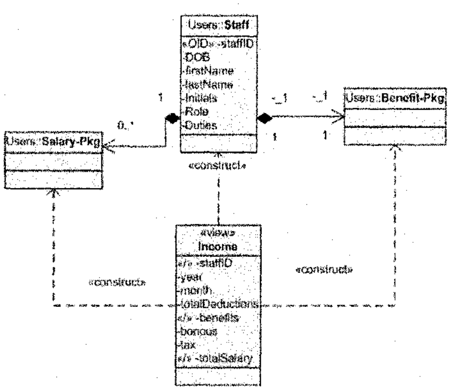


Figure 3. A conceptual view example (Income)

Example 14: A compression of elements indicates that those elements are replaced by a single element in the Ontology view [14]. The element itself can be a new element, but it will not provide additional semantic information (compared to the base ontology). The compression operator constituted of one or more of unary operations combined in sequence.

7. CONCLUSION AND FUTURE WORK

Views have proven to be very useful in databases and here, we presented a descriptive discussion of an abstract view model for SW (SW-view). We first provided formal properties of the SW-view model including a set of binary and unary conceptual operators. Secondly, we provided a brief discussion on issues related to SW-view model, including some modelling issues and the view constraint model. Then we briefly presented how SW-views can be utilized in the MOVE system, followed by some illustrative SW-view based on an industrial case study.

For future work, some further issues deserve investigation. First, the investigation of a formal mapping and transformation approach of the view constraints, and to automate the constraint model transformation between the SW-view model to SW languages, such as RDF and OWL schema constraints. Second, the automation of the mapping process between conceptual operators to various SW (high-level) query language expressions (e.g. RDQL) with emphasis on performance. Third, is the investigation into the dynamic properties of the SW-view model.

8. REFERENCES

[1] C. Wouters, T. S. Dillon, J. W. Rahayu, E. Chang, and R. Meersman, "Ontologies on the MOVE," 9th International Conference on Database Systems for Advanced Applications (DASFAA '04), Jeju Island, Korea, 2004.

[2] C. Wouters, T. S. Dillon, J. W. Rahayu, E. Chang, and R. Meersman, "A Practical Approach to the Derivation of a Materialized Ontology View," in *Web Information Systems, Edited by D. Taniar and J. W. Rahayu*, D. Taniar and W. Rahayu, Eds. USA: Idea Group Publishing, 2004.

[3] R. Volz, D. Oberle, and R. Studer, "Views for light-weight Web ontologies," Proceedings of the ACM Symposium on Applied Computing (SAC '03), USA, 2003.

[4] R. Elmasri and S. Navathe, *Fundamentals of database systems*, 4th ed. New York: Pearson/Addison Wesley, 2004.

[5] W. Kim and W. Kelly, "Chapter 6: On View Support in Object-Oriented Database Systems," in *Modern Database Systems*: Addison-Wesley Publishing Company, 1995, pp. 108-129.

[6] P. Spyns, R. Meersman, and J. Mustafa, "Data Modeling Versus Ontology Engineering," SIGMOD, 2002.

[7] T. S. Dillon and P. L. Tan, *Object-Oriented Conceptual Modeling*: Prentice Hall, Australia, 1993.

[8] I. Graham, A. C. Wills, and A. J. O'Callaghan, *Object-oriented methods : principles & practice*, 3rd ed. Harlow: Addison-Wesley, 2001.

[9] OMG-MDA, "The Architecture of Choice for a Changing World®, MDA Guide Version 1.0.1 (http://www.omg.org/mda/)," OMG, 2003.

[10] L. Feng, E. Chang, and T. S. Dillon, "A Semantic Network-based Design Methodology for XML Documents," *ACM Transactions on Information Systems (TOIS)*, vol. 20, No 4, pp. 390 - 421, 2002.

[11] R.Rajugan, E. Chang, L. Feng, and T. S. Dillon, "Semantic Modelling of e-Solutions Using a View Formalism with Conceptual & Logical Extensions," 3rd International IEEE Conference on Industrial Informatics (INDIN '05), Perth, Australia, 2005.

[12] OMG-UML™, "UML 2.0 Final Adopted Specification (http://www.uml.org/#UML2.0)," 2003.

[13] W3C-OWL, "OWL: Web Ontology Language 1.0 reference (http://www.w3.org/2004/OWL/)," W3C, 2004.

[14] C. Wouters, Rajugan R., T. S. Dillon, and J. W. Rahayu, "Ontology Extraction Using Views for Semantic Web," in *Web Semantics and Ontology*, D. Taniar and W. Rahayu, Eds. USA: Idea Group Publishing, 2005.

[15] C. Wouters, T. S. Dillon, J. W. Rahayu, and E. Chang, "A Practical Walkthrough of the Ontology Derivation Rules," Database and Expert Systems Applications : 13th International Conference (DEXA '02), Aix-en-Provence, France, 2002.

[16] W3C-OWL, "OWL web ontology language 1.0 reference," W3C, 2002.

[17] W3C-RDF, "Resource Description Framework (RDF), (http://www.w3.org/RDF/)," 3 ed: The World Wide Web Consortium (W3C), 2004.

[18] I. Cruz, S. Decker, J. Euzenat, and D. McGuinness, "The emerging semantic web : selected papers from the first Semantic Web Working Symposium," IOS Press,Tokyo : Ohmsha, 2002, pp. 297.

[19] P. Wongthamtham, E. Chang, T. S. Dillon, J. Davis, and N. Jayaratna, "Ontology Based Solution for Software Development," International Conference on Software Engineering and Applications (ICSSEA '03), Paris, France, 2003.

[20] D. Gaševic, D. Djuric, V. Devedzic, and V. Damjanovic, "Approaching OWL and MDA Through Technological Spaces," 3rd Workshop in Software Model Engineering (WiSME 2004), Lisbon, Portugal, 2004.

[21] D. Gaševic, D. Djuric, V. Devedzic, and V. Damjanovic, "Converting UML to OWL Ontologies," Proceedings of the 13 th International World Wide Web Conference, NY, USA, 2004.

[22] C. J. Date, *An introduction to database systems*, 8th ed. New York: Pearson/Addison Wesley, 2003.

[23] S. Abiteboul and A. Bonner, "Objects and Views," ACM SIGMOD Record, Proceedings of the International Conference on Management of Data (ACM SIGMOD '91), 1991.

[24] S. Abiteboul, "On Views and XML," Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS '99), Philadelphia, Pennsylvania, USA, 1999.

[25] Y. B. Chen, T. W. Ling, and M. L. Lee, "Designing Valid XML Views," Proceedings of the 21st International Conference on Conceptual Modeling (ER '02), Tampere, Finland, 2002.

[26] R.Rajugan, E. Chang, T. S. Dillon, and F. Ling, "A Three-Layered XML View Model: A Practical Approach," 24th International Conference on Conceptual Modeling (ER '05), Klagenfurt, Austria, 2005.

[27] W3C-XML, "Extensible Markup Language (XML) 1.0, (http://www.w3.org/XML/)," 3 ed: The World Wide Web Consortium (W3C), 2004.

[28] Y. Zhuge and H. Garcia-Molina, "Graph structured Views and Incremental Maintenance," Proceeding of the 14th IEEE Conference on Data Engineering (ICDE '98), USA, 1998.

[29] S. Abiteboul, R. Goldman, J. McHugh, V. Vassalos, and Y. Zhuge, "Views for Semistructured Data," Workshop on Management of Semistructured Data, USA, 1997.

[30] H. Liefke and S. Davidson, "View Maintenance for Hierarchical Semistructured," Proceedings of the Second International Conference on Data Warehousing and Knowledge Discovery (DaWak '00), London, UK, 2000.

[31] R.Rajugan, E. Chang, T. S. Dillon, and F. Ling, "XML Views: Part 1," 14th International Conference on Database and Expert Systems Applications (DEXA '03), Prague, Czech Republic, 2003.

[32] S. Cluet, P. Veltri, and D. Vodislav, "Views in a Large Scale XML Repository," Proceedings of the 27th VLDB Conference (VLDB '01), Roma, Italy, 2001.

[33] R. G. G. Cattell, D. K. Barry, M. Berler, J. Eastman, D. Jordan, C. Russell, O. Schadow, T. Stanienda, and F. Velez, "The Object Data Standard: ODMG 3.0," Morgan Kaufmann, 2000, pp. 300.

[34] Xyleme, "Xyleme Project (http://www.xyleme.com/)," 2001.

[35] Lucie-Xyleme, "Lucie Xyleme: A dynamic warehouse for XML Data of the Web," *IEEE Data Engineering Bulletin*, vol. 24, No 2, pp. 40-47, 2001.

[36] B. Ludaescher, Y. Papakonstantinou, P. Velikhov, and V. Vianu, "View Definition and DTD Inference for XML," Post-ICDT Workshop on Query Processing for Semistructured Data and Non-Standard Data Formats, 1999.

[37] R.Rajugan, E. Chang, T. S. Dillon, and F. Ling, "A Layered View Model for XML Repositories & XML Data Warehouses," The 5th International Conference on Computer and Information Technology (CIT '05), Shanghai, China, 2005.

[38] W3C-SW, "Semantic Web, (http://www.w3.org/2001/sw/)," W3C, 2005.

[39] R. Volz, D. Oberle, and R. Studer, "Implementing Views for Light-Weight Web Ontologies," Seventh International Database Engineering and Applications Symposium (IDEAS'03), Hong Kong, SAR, 2003.

[40] KAON, "KAON Project (http://kaon.semanticweb.org/Members/rvo/Folder.2002-08-22.1409/Module.2002-08-22.1426/view)," 2004.

[41] L. Feng, E. Chang, and T. S. Dillon, "Schemata Transformation of Object-Oriented Conceptual Models to XML," *International Journal of Computer Systems Science & Engineering*, vol. 18, No. 1, pp. 45-60, 2003.

[42] C. Wouters, T. S. Dillon, J. W. Rahayu, E. Chang, and R. Meersman, "A Practical Approach to the Derivation of a Materialized Ontology View," in *Web Information Systems, Edited by D. Taniar and J.W. Rahayu, Idea Group Publishing, USA, 2004*, D. Taniar and W. Rahayu, Eds. USA: Idea Group Publishing, 2004.

[43] HyOntUse, "User Oriented Hybrid Ontology Development Environments, (http://www.cs.man.ac.uk/mig/projects/current/hyontuse/)," 2003.

[44] L. Feng, T. S. Dillon, H. Weigand, and E. Chang, "An XML-Enabled Association Rule Framework," 14th International Conference on Database and Expert Systems Applications (DEXA '03) 2003, Prague, Czech Republic, 2003.

[45] R. Conrad, D. Scheffner, and J. C. Freytag, "XML conceptual modeling using UML," 19th International Conference on Conceptual Modeling (ER '00), USA, 2000.

[46] D. Gaševic, D. Djuric, V. Devedžic, and V. Damjanovic, "UML for Read-To-Use OWL Ontologies," Proceedings of the IEEE International Conference Intelligent Systems, Vrana, Bulgaria, 2004.

[47] M. Golfarelli, D. Maio, and S. Rizzi, "The Dimensional Fact Model: A Conceptual Model for Data Warehouses," *International Journal of Cooperative Information Systems*, vol. 7, pp. 215-247, 1998.

[48] J. Trujillo, M. Palomar, J. Gomez, and I.-Y. Song, "Designing Data Warehouses with OO Conceptual Models," in *IEEE Computer Society, "Computer"*, 2001, pp. 66-75.

[49] W3C-XPath, "XML Path Language (XPath) Version 1.0," in *XML Path Language*, vol. November 1999: The World Wide Web Consortium (W3C), 1999.

[50] S. Abiteboul, J. Quass, J. McHugh, J. Widom, and J. Wiener, "The Lorel Query Language for Semistructured Data," *International Journal on Digital Libraries*, vol. 1, pp. 68-88, 1997.

[51] R. Goldman, J. McHugh, and J. Widom, "From Semistructured Data to XML: Migrating the Lore Data Model and Query Language," Proceedings of the 2nd International Workshop on the Web and Databases (WebDB '99), Philadelphia, Pennsylvania, 1999.

[52] OMG-OCL, "UML 2.0 OCL Final Adopted specification (http://www.omg.org/cgi-bin/doc?ptc/2003-10-14)," OMG, 2003.

[53] H. Balsters, "Modelling Database Views with Derived Classes in the UML/OCL-framework," The Unified Modeling Language: Modeling Languages and Applications (UML '03), USA, 2003.

[54] J. B. Warmer and A. G. Kleppe, *The object constraint language : getting your models ready for MDA*, 2nd ed. Boston, MA: Addison-Wesley, 2003.

[55] E. Chang, T. S. Dillon, W. Gardner, A. Talevski, Rajugan R., and T. Kapnoullas, "A Virtual Logistics Network and an e-Hub as a Competitive Approach for Small to Medium Size Companies," 2nd International Human.Society@Internet Conference, Seoul, Korea, 2003.

[56] E. Chang, W. Gardner, A. Talevski, E. Gautama, Rajugan R., T. Kapnoullas, and S. Satter, "Virtual Collaborative Logistics and B2B e-Commerce," e-Business Conference, Duxon Wellington, NZ, 2001.

[57] ITEC, "iPower Logistics (http://www.logistics.cbs.curtin.edu.au/)," 2002.

INFORMATION ARCHITECTURES FOR SEMANTIC WEB APPLICATIONS

Kimmo Salmenjoki[1], Yaroslav Tsaruk[1], Gurusamy Arumugam[2]
[1]*University of Vaasa, Box 700, 65101 Vaasa, Finland* , [2]*Madurai Kamaraj University, Madurai-625021, Tamilnadu, India*

Abstract: In this paper we discuss the issues related to information system technologies. The problems of hardware based services and insufficiency in the widely accepted standards have limited dynamic aspects of industrial applications. Besides the hardware, software and communication considerations one has to look into the system integration, user involvement and service aspects due to the impact of web technologies. We present some of the high level approaches provided by the semantic web technologies in a heterogeneous collaboration network and also discuss the approaches for designing and describing the information architecture of these information systems.

Key words: semantic web; XML; RDF; ontologies; information architectures; software services

1. INTRODUCTION

The innovations in information system (IS) technologies and IT vendor products have been the technological driver in the evolution of industrial applications. The present practices in industrial applications are a colourful collection of different approaches mostly unified by networked TCP/IP only at low level data granularity. To be more service oriented we have to see simultaneously both hardware and software coherently on the organisation of industrial information systems (IIS).

Due to heterogeneous environments the dynamic aspects of industrial applications have been strictly limited. Integration of different data sources and devices has always needed further software development. When different system components need to be integrated either from the business or process point of view, higher level approaches other than the network and data based integration also have to be addressed. In combining hardware and software, the key issue in information system development has previously been data communication, but the recent penetration of web technologies has shifted the focus towards system integration, user involvement and service and lifecycle aspects. In most cases now-a-days one is eager to consider the life time of industrial product and services which again sets more demand on the abstraction level of software and data for system integration.

The work of Industrial Ontologies Group (IOG) in the University of Jyväskylä, Finland, focuses on establishing industry wide approaches of sharing knowledge and building industrial applications that contain also autonomous and self learning components. Their work on the project Smart Resources has already proved the dire need of new approaches to provide sustainable unifying industrial services for business customers. Related to this wider approach we will describe a case example of applying IOG approaches towards industrial projects containing several actors, see [1, 2].

In what follows, we will go through high level approaches provided by the semantic web technologies in a case scenario taking into account the heterogeneous collaboration network. This paper will address approaches for combining logic and data by using the semantic web based technologies. The case that we present for this approach is figurative, as we describe educational units and their operations from the mutual collaboration point of view, but the approach we suggest should be also valid for other IS development cases.

2. SOLUTION FOR REFINED SOFTWARE AND INFORMATION ARCHITECTURES IN INFORMATION SYSTEMS

2.1 Service based application development in IIS with web technologies

In this chapter we will address the methodologies that have been used for making coherent approaches in both software and data sharing in information systems. Different component approaches like DCOM or CORBA have been successfully used only on the E-business related applications but not on the IIS due to the existence of incoherent industrial environments. This component based approach is only addressing the heterogeneous software dilemma leaving many of the data related dependencies to the very low level in the network.

The developments in web environment is also penetrating the IS applications either via XML features provided by the database vendors or through other higher level data description. More or less in the near future IS development can be seen from two different approaches: either consisting of unified services or shared data. Again the already existing heterogeneous legacy approaches will be an obstacle for the wider applicability of web services to produce more unified shared logic among the individual software industrial application systems. At least this has been the case for the last ten years, though the components have gained wide acceptability in the enterprise and IT vendor communities.

The implicit usage of XML makes the technical exchange of data more transparent. Partly it also makes data visible beyond application boundaries. The wider usage of web services as a basis of software development still requires deeper consideration of software architecture, see [3].

The previously mentioned components approaches have been unified to a new approach of building software with web service components that have been standardized by W3C. This means, in practice, XML and its technologies have become the de facto approach for sharing data both within and between applications. Again on the enterprise side all IT vendors are strongly pushing and promoting the usage of web services as a methodological approach to develop information systems and improve their integration properties in the future. The conservativeness of industrial application will slow these developments in IS. However the thorough penetration of IT in all modern systems implies these developments also in the industrial side.

These crossovers also introduce some problems on the overall architecture and management of system components as possible services with accurately described components and data interfaces. The previously described software architecture does not necessarily account for all the practical implications of the complex applications as distributed applications, see [4].

This discussion has further lead into service oriented architectures and its future application in software factory type manner, see [5]. This approach fits well with the previously described service approach for IS development although it, at the same time, highly complicates the manageability, security and interoperability issues by making the software components again more granular.

2.2 Applying semantic web approach for software services

In the classical web environment the semantic web approach has been mostly developed to contain shared information architecture within some restricted application domains, most notably the library information systems and their organisation. After dozens of years of development in both information description and IT development, especially information data bases and encoding, the global community of library developers have come up, and widely implanted, the usage of Dublin core (DC) as the shared abstract data description in the libraries all over the world. At the present, however, the web technology improvements will question the practical expandability of the Dublin Core to other application domains. For any application domain of semantic web this process should be a reminder of the complexity in developing widely shared vocabularies and their sharing.

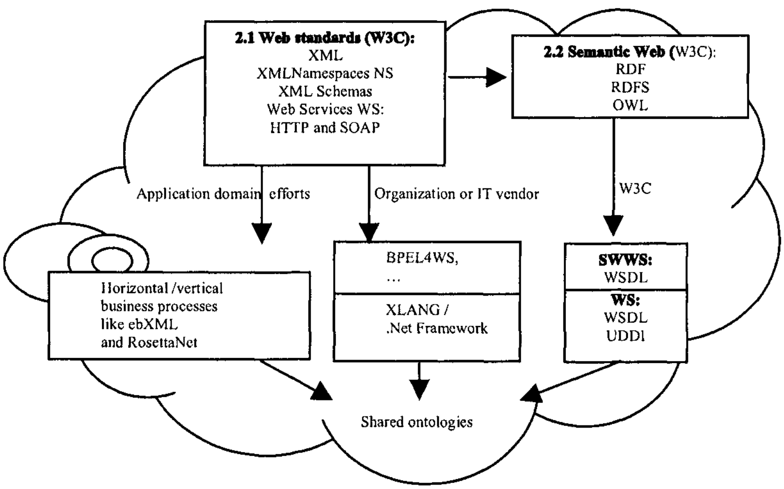


Figure 1. Four different approaches for unified service based IS development

In Figure 1 the three different technological approaches for building service based IS are given by the three named arrows. The technical specifications and explanations for the respective standards and efforts are given in [6, 7]. The fourth, shared ontologies approach, is represented by the cloud that contains all the technological approaches used in a coherent way. The introduction of semantic web eases the software complexity, at least in the enterprise and web scenarios, and also improves the previously discussed low level granularity of the data layer. After the presently on going standardisation process and its acceptance both by the IT vendors and the industrial developers the necessity of unifying software and data structures is the simultaneous application of semantic web and web services. When the IT software development tools become XML and web services enabled the next higher approaches for information architecture will be based on the RDF, RDFS and domain space specific ontologies.

Using RDF can also be seen as a way of building self-aware and proactive data. This is the need put forward by the two previously described views of IS software consisting either from unified services or shared data. Assuming that the semantic web environment tools are available, an essential part of the complexities of both software and information architecture could be addressed in the XML and RDF description of processes as services. In general, from the services and structural point of view the vendor specific versions are the preferred ones, whereas from the operational and standardization point of view the low level granularity approach prevails. In a unified development environment business processes can also be shared in integrated manner. From the previously described web services approach the methodological interest in new standardisation has moved towards distributed computing in heterogeneous environments, see [8]. As an example of the unification that is presently happening, we consider next an example case of the educational domain and it's information architecture. Addressing both software development (Section 2.1) and semantic web aspects (Section 2.2) we address the decentralized software architecture dilemmas.

3. APPROACH FOR INFORMATION DESIGN WITH SEMANTIC WEB-AN EXAMPLE

3.1 MODE approach for analyzing Baltic Sea Network BSN project data

In the University of Vaasa, we are running the project MODE, which addresses Management Of Distributed Expertise. In this project we have analyzed different cases, where several networked organisations share interest and knowledge on common projects. Although educational units collaborate continuously, there are many problems to establish common terminologies among the universities or while working in specific projects as all things are heavily language, culture and practical operational habits.

To simplify the handling, we next introduce the Baltic Sea Network BSN as a case project of MODE and discuss its information architecture in detail. The main purpose of BSN is to combine efforts in sharing education and research operations and interests among the partner universities. The network promotes international co-operation focusing on the following areas: Welfare, Business Skills and Management, Tourism, and Information and Communication Technology always taking into account the sustainable development. To the BSN network belongs about 40 educational organisations. Each of these organisations has a list of courses, which could be available to any of student of BSN institution. We use BSN, as a case project of MODE, to create globally sharable structure data schema for the communication of data among participating organizations. The general project related information architecture will be left out of this paper.

Using the general methodologies in the Figure 2 we will next give a design for an information architecture that encapsulates the BSN project metadata in the educational application domain. As the general approach of Figure 1 is still somewhat unclear, we give here an example of the educational domain shared ontology development process.

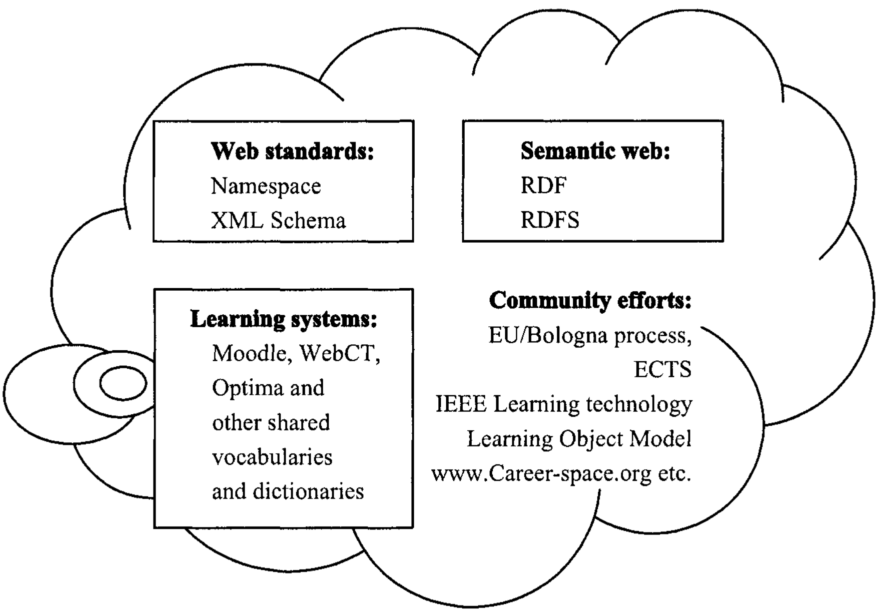


Figure 2. Options for metadata placement in information architecture of the educational domain

As described before for generic IS and software development in previous chapter, here we can use web standard or semantic web approaches. In the educational domain the three arrows in the middle of Figure 1 are cut down into two, as the standardizing organizations self contain education suppliers i.e. universities. Like before, also here we have to consider the granularity and access of the specific metadata items. As we see from the Figure 2, in the educational domain, the semantic web has not yet gained any popularity, although web pages and tools have been widely used for delivering information, personal and group based communication in education.

Here we can say that as we are using the standard semantic web approach, all the standards and tools of semantic web will be available to produce the knowledge aware applications for the educational domain. The suggested information design has to correspond to the needs of the semantic web applications that will be developed for the project and educational domain in Figure 2. Besides the classical search engine type applications we will also produce web based information gathering applications that are dynamically linked to the respective home pages of the partner universities of Baltic Sea Network. The semantic web based applications enable intensifying collaboration in specific subject areas. On the research side, semantic web application will be built to link the suggested project proposals (with its tenders) to partners working in the same interest area. Internally each partner can also use this information for internal resource planning. For the information design purposes we will next give more technical approaches to design and describe the BSN project metadata.

3.2 Technical description of the information contained in the BSN

At present the BSN has a website and the Optima learning environment is used for communicating data and information exchange among the partners. Based on the approaches in Figure 2, we will describe the generic educational organization data structure next. The overall information structure is given in Figure 3.

The whole structure presents information about educational units, personnel and courses. The information about an educational unit is presented by classes such as "Organisation" and "Department". The information about a staff in the organisation is given by the classes "Person", "Staff", "Degree". Class "Person" contains basic information about any one working in the university. The "Degree" class includes data about educational qualifications of a staff working in the university by the attribute "state" in that class. The class "Staff" relates the person and degree information.

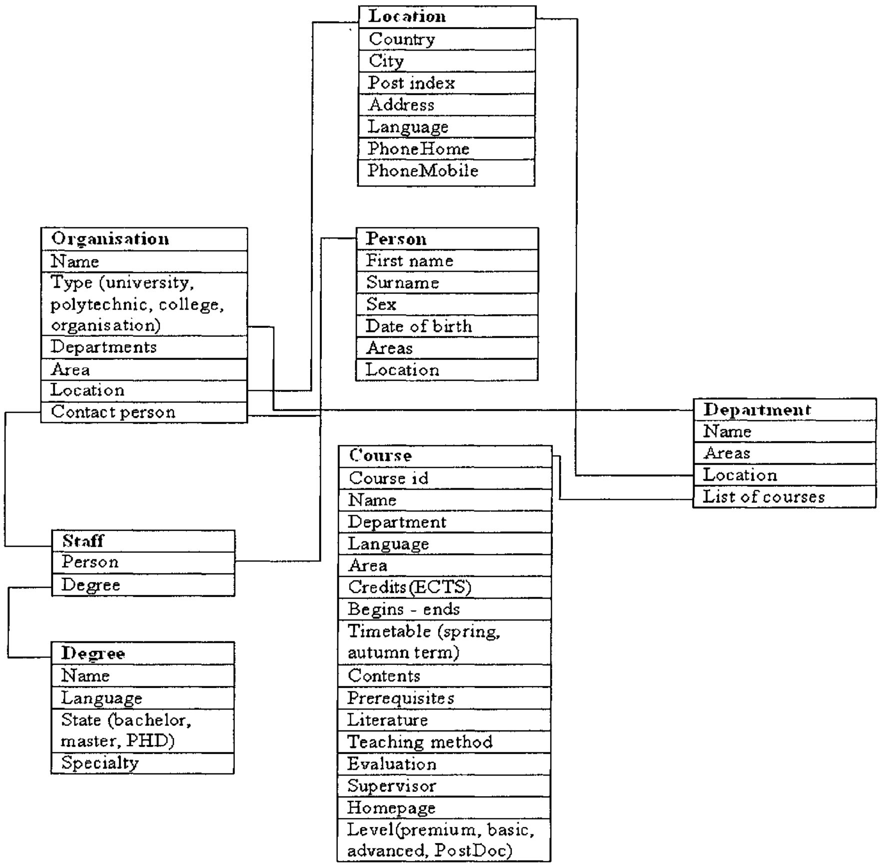


Figure 3. The structure of information contained in BSN project

The structure presented in Figure 3 is next communicated within the network and refined further against the higher level community standards of Figure 2. We next present, in Example listings 1 and 2, the metadata structures related to XML and RDF descriptions for some items of Figure 3.

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!-- Namespace Declarations in XMLSchema -->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
        xmlns:edu = "http://people.jyu.fi/~yatsaruk" version = "1.0">
```

```
<xsd:complexType name = "Organisation">
   <xsd:sequence>
     <xsd:element name = "name" type = "xsd:string" />
     <xsd:element name = "type" >
    <xsd:simpleType>
        <xsd:restriction base = "xsd:string">
         <xsd:enumeration value = "University"/>
         <xsd:enumeration value = "Polytechnic"/>
     <xsd:enumeration value = "College"/>
        </xsd:restriction>
    </xsd:simpleType>
   </xsd:element>
  <xsd:element name="Departments" />
  <xsd:sequence>
     <xsd:element name = "Domain" type = "xsd:string" />
     <xsd:element ref="edu:Location" />
     <xsd:element ref="edu:Staff" />
   </xsd:sequence>
</xsd:complexType>
```

Organisatio definition

```
<xsd:complexType name = "Department">
   <xsd:sequence>
     <xsd:element name = "Name" type = "xsd:string" />
     <xsd:element name = "Domain" type = "xsd:string" />
     <xsd:element ref="edu:Location" />
     <xsd:element name="Courses" />
    <xsd:sequence>
         <xsd:element ref="edu:Course" />
    </xsd:sequence>
   </xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

Department definition

Example 1. Metadata presented in XML Schema format

Description of Organisation and Department is given in Example 1. Both these units are presented as complexType elements. Each of these complexType elements includes a set of elements. The element could be of a basic type or another complex element. The "Type" property is declared as enumerated one. The field location is defined by reference to complex element Location, situated in "edu" namespace. The "Department" unit contains element such as “Courses”, which describes the list of courses proposed by this department. The element “Courses” is a set of reference to element “Course”, which is situated in “edu” namespace.

On specific application domains the IT vendors will be gradually improving their information granularity in a manner similar to what Microsoft has done on the IT applications and their end users side. In the simple office and enterprise scenario some vendors like Microsoft have provided examples of unified approaches like the .NET Framework as a distributed computing environment or unifications of information either by application to application data sharing or usage of XML namespace based information exchange like sharing data between Microsoft Office applications via the Microsoft Office Namespace Schema. An example domain, where this has been widely used, is the Danish national effort to unify communal system and processes developments by sharing the MS-Office namespaces between different communal actors and applications, see [9]. In that case this makes the MS-Office namespace as the universal information sharing architecture among any information exchange partners. This is the lowest level of creating information architectures which is based notably on shared XML namespace usage.

```
<rdfs:Class rdf:ID="Staff">
 <rdfs:comment>Staff Class</rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://people.jyu.fi/~yatsaruk/world#Person"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Organisation">
 <rdfs:comment>Class for general organization units</rdfs:comment>
</rdfs:Class>
<rdfs:Class rdf:ID="EducationalUnit">
 <rdfs:comment>Class for educational organization units</rdfs:comment>
 <rdfs:subClassOf rdf:resource="#Organisation"/>
</rdfs:Class>
<rdfs:Class rdf:ID="University">
 <rdfs:comment>Class for universities</rdfs:comment>
 <rdfs:subClassOf rdf:resource="#EducationalUnit"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Polytechnic">
 <rdfs:comment>Class for polytechnics</rdfs:comment>
 <rdfs:subClassOf rdf:resource="#EducationalUnit"/>
</rdfs:Class>
<rdfs:Class rdf:ID="College">
 <rdfs:comment>Class for colleges</rdfs:comment>
 <rdfs:subClassOf rdf:resource="#EducationalUnit"/>
```

```
</rdfs:Class>
<rdfs:Class rdf:ID="Department">
 <rdfs:comment>Class for departments</rdfs:comment>
 <rdfs:subClassOf rdf:resource="#EducationalUnit"/>
</rdfs:Class>
<rdf:Property rdf:ID="partOf">
 <rdfs:comment>Part-of relationship</rdfs:comment>
 <rdfs:domain rdf:resource="#Organisation"/>
 <rdfs:range rdf:resource="#Organisation"/>
</rdf:Property>
 <rdf:Property rdf:ID="name">
  <rdfs:comment>The name of organisation or department</rdfs:comment>
  <rdfs:domain rdf:resource="#Organisation"/>
 </rdf:Property>
<rdf:Property rdf:ID="Location">
  <rdfs:comment>Location of organisation</rdfs:comment>
  <rdfs:domain rdf:resource="#Organisation"/>
</rdf:Property>
<rdf:Property rdf:ID="ContactPerson">
  <rdfs:comment>Contact person of organisation</rdfs:comment>
  <rdfs:domain rdf:resource="#Organisation"/>
  <rdfs:range rdf:resource="#Staff"/>
</rdf:Property>
</rdf:RDF>
```

Example 2. Presentation of metadata structure in RDFS format

The information in RDF format is presented by two elements: classes and properties. The definition of classes presented in the beginning of RDF document. The "Organisation" class is inherited from "Resource". The parents for class "Staff" is "Person". The description of properties depicted in the rest of RDF document. The property type Alt is used for describing container where just one option should be selected of the values attributed to the element in the RDF.

All information exchanges that take place in the previously described BSN case project can be seen from two different views: internally from the organization or externally from the BSN project point of view. In internal is a data presentation format used inside each institution, and this is reflected on the web pages of the respective universities. These sets of data characteristics for each educational organisation are specified in their own way. After structuring and presenting this metadata in RDF format it becomes external format of the BSN data. This data and its metadata will be used to enhance the information sharing between the partners and further to describe knowledge based applications.

3.3 Comparing XML and RDF approaches for BSN information architecture

Next we will clarify in detail the advantages and disadvantages of using the XML and RDF based approaches in Example 1 and Example 2.

The examples discussed before show the different presentation formats of the same information. The structuring and presentation of information are different in both formats but namespaces have been used in both of them. In XML format namespaces need not point to anything in the XML Namespace specification. In RDF, the namespace URI reference also identifies the location of the RDF schema. RDF format presents object oriented paradigm. Resource is the top level class. Latest revisions to the RDF specifications allow cycles in class hierarchy which was not there earlier. In XML the information is presented by elements of certain Type. The type can be of two types: simple and complex. Complex contains a set of elements inside. The class is defined by elements and their properties could be another element. It has no defined semantics. Inheritance can not been realized in XML format. However, types can be “extended” or “restricted”, thus defining subTypes. But in RDF along with object and classes inheritance can also be realized. A class can be a subClassof other classes (multiple inheritance is allowed). The inheritance is related to Property. Properties can be subPropertyOf other properties. The type of data used in XML and RDF formats is different. In RDF the core RDF Schema includes “Literals” which is the set of all strings. The latest RDF specification is expected to include XML Schema data types. The data types supported by XML Schema are mainly variations of numerical, temporal and string data types. The XML format allows describing the enumeration of properties using <enumeration> tag. The RDF doesn’t allow such possibility, see [10].

Finally the essential question is on the users of the information architectures and also on the necessary applications that would be using this architecture. In case of the learning systems the evolution of the systems is presently covering the XML as medium of sharable information. Also here the IT vendor supported technologies and tools, most notably the web services, are the likely interfaces that will give access to the granularly refined learning objects, which are shared by the learning communities. Again also here, with the simultaneous usage of semantic web to provide meaning and web services to provide the access, we are able to unify the learning objects into knowledge based educational applications.

4. CONCLUSIONS

We have seen in this paper how the information design is a necessity for building knowledge based applications. Once the information architecture is given then the general methodologies, technologies and related tools can easily enable knowledge based applications in the domain scope. When refining the project metadata in general we will split the metadata into different sections like strategic tasks, human resources and contextual connections. Also here the ontological approach will be combined to the work of Mikko Laukkanen and Heikki Helin, who have built semantic web applications for finding an expert within an organisation [11]. When building the higher part of the ontologies the European Union wide curricula and degree structures would provide models for the sharable ontologies.

As next step in our approach, we will refine the information extraction phase in the Figure 1, so that we can automatically harvest as much of the above data, related to both the partners and their educational offering. We will consider the semantic web software and application needs of this case in more detail in the follow up papers [12, 13].

REFERENCES

1. Terziyan V., Semantic Web Services for Smart Devices in a Global Understanding Environment in: *On the Move to Meaningful Internet Systems 2003*: edited by Meersman R. and Tari Z. (OTM 2003 workshops, Lecture notes in Computer Science, Vol.2889, Springer-Verlag, 2003, pp.279-291)
2. Kaykova O., Khriyenko O., Kovtun D., Naumenko A., Terziyan V., Zharko A., General Adaptation Framework: Enabling Interoperability for Industrial Web Resources, *International Journal on Semantic Web and Information Systems*, Idea Group, ISSN:1552-6283, Vol.1 No.3, July-September 2005 pp.30-62 (to appear)
3. Alesso H.P., Smith C.F: *The Intelligent Wireless Web*, Addison Wesley, Pearson Education, 2002.
4. Singh M.P, Huhns M.V., *Service Oriented Computing: Semantics, Processes, Agents,* Wiley, 2004.
5. Greenfield J., Short K.: *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*, Wiley, 2004.
6. World Wide Web Consortium, Semantic Web, http://www.w3.org/2001/sw/

7. World Wide Web Consortium, Web Services, http://www.w3.org/2002/ws/
8. Da Conta M.C., Obrst L. J., Smith K.T.: The *Semantic Web: A guide to the Future of XML, Web Services and Knowledge Management*, Wiley 2003
9. Danish e-Governance Offentlig Information Online, http://www.oio.dk/English
10. Gil Y. and Ratnakar V. "A Comparison of (Semantic) Markup Languages" USC Information Sciences Institute and Computer Science Department; http://trellis.semanticweb.org/expect/web/-semanticweb/paper.pdf
11. Laukkanen M. Helin H., Competence manager- applying semantic web in practice in: *Third International Semantic Web Conference (ISWC2004), Proceedings Series: Lecture Notes in Computer Science, Vol. 3298*, edited by McIllraith S., Plexsousakis D., Harmelen F. van; http://iswc2004.semanticweb.org/demos/22/paper.pdf
12. Salmenjoki K., Terziyan V., Approaching Web Based Information With Semantic Web, Unpublished Manuscript
13. Tsaruk Y., Salmenjoki K., Project Casing Pilot System., Unpublished Manuscript

SECURING WEB SERVICES USING SEMANTIC WEB TECHNOLOGIES

Brian Shields, Owen Molloy, Gerard Lyons and Jim Duggan
Department of Information Technology,
National University of Ireland,
Galway, Ireland.
brian.shields@geminga.it.nuigalway.ie, {owen.molloy,gerard.lyons,jim.duggan}@nuigalway.ie

Abstract In this paper we propose an authorisation definition and access control solution for Web Services. In our proposal we define our access control policies using an OWL-DL language based on the eXtensible Access Control Markup Language (XACML). We propose the use of resource and subject metadata ontologies, also written in OWL. We then present a complete Web Services architecture which incorporates this access control model. As part of this architecture we propose a novel document filtering mechanism according to the semantic enriched access control policies.

Keywords: Web Services Security, XACML, OWL, Semantic Web, Access Control

1. Introduction

The World Wide Web is growing at a exponential rate [1]. There are more and more technologies being developed to provide different ways of accessing this huge information resource, as well as representing the information stored. Because of the increase in information available and of people or agents accessing it, the issue of securing this information has become paramount.

Access control is the current 'hot topic' in information security. It has become necessary to define security policies that allow a person describe who can access what information, where they access it from, when they can access it and how they can access it. However, as each new item of information is added to this secure environment, is it necessary to define all of these policy issues again? Does each new user have to be added to all of these access lists? Unfortunately until recently the answer was yes. There have been strides taken in the advancement of security policies on the Web. Role Based Access Control has become commonplace, where a new user can be granted all the permissions of a particular role or group.

This paper will focus on access control for the Web Services environment and how these access control practices can be improved by enriching them with machine processable semantics. Therefore not only can we define users as part of groups or roles, but we can now group stored information according to its meaning. This paper proposes a way of augmenting existing Web Service access control standards to become semantically-aware. We define a security architecture which incorporates this novel access control policy. We provide a novel document filtering algorithm according to access control policies, therefore not limiting access control to a document to a boolean value.

In the next section we discuss the two principal standards which will be used from both the Web Services security field and the Semantic Web arena. We then present our solution which includes a proposed policy language, limited document access algorithm and a security architecture which encompasses these. We have included a section on related research in this field and present our conclusions.

2. Technologies

It is important when developing a solution to be aware of standards and standard practices in the area. For the solution we propose in this paper we straddle two key areas of todays World Wide Web research: Web Services security and the Semantic Web. From the standards in Web Services security we focus on XACML. The Semantic Web technology used is OWL. This section introduces these standards.

2.1 XACML

XACML or eXtensible Access Control Markup Language is defined by the OASIS standards body [2]. XACML is an XML based language used to construct access control policies for Web Services environments. XACML can grant or refuse access to protected resources based on attributes of the requester, the protocol used to access the resource, authentication methods or even global settings such as time of day and location.

There are six principal components of the XACML architecture:

- **PEP**, or Policy Enforcement Point, intercepts a SOAP request and constructs a SAML authorisation decision query from the information in the request. Rules pertaining to the access query, while they may be defined and evaluated elsewhere, are enforced here.
- **PDP**, or Policy Decision Point, evaluates the rule or rules in the policy. The policy can be retrieved from the PRP if it is not cached on the PDP. Once the policy has been evaluated, a SAML authorisation decision assertion is returned to the PEP.

- **PRP**, or Policy Retrieval Point, returns the requested policy to the PDP. If the policy is not available at the PRP it can be retrieved from the policy store. This will be the case if the policy has never been requested before, or if the policy is being refreshed.
- **PIP**, or Policy Information Point, is used to calculate the predicate of a rule. In XACML, the predicate is defined as "the ability to query an attribute" [2]. The attribute information is returned to the PRP, from where it was requested, in the form of a SAML attribute assertion.
- **PAP**, or Policy Administration Point, creates rules, combines rules into policies, and uploads these policies to the policy store. The PAP usually takes the form of a graphical console and uploads the policies as XACML.
- **Policy store** is used to store the rules and policies defined at the PAP. While XACML is used for the import and export of policies, they are not necessarily stored in their native format. They may be stored in a traditional relational database with an XML interface.

There are three essential parts to XACML policy writing:

- **Rule:** A rule is the most elementary piece of a policy. Rules are encapsulated inside a policy. Rules are evaluated based on their contents. The main components of a rule are a target, which is a set of resources, subjects or actions to which the rule is intended to apply; an effect, which contains the rule writers intended consequence if the rule is evaluated to be true; and a set of conditions.
- **Policy:** According to O'Neill [3], this is "perhaps the most important aspect of the specification". A policy contains a set of rules, an algorithm for combining these rules, a target, similar to that in rule, and a set of obligations. An obligation is defined as an action to be performed once the authorization decision is complete.
- **PolicySet:** This is a set of policy elements, a policy combining algorithm, a target and a set of obligations.

2.2 OWL

OWL [4] was created by the W3C Web Ontology working group. It is based on the DAML+OIL language, and is layered on top of RDF and RDFS. OWL actually consists of three sub languages or dialects: OWL Lite, OWL DL and OWL Full. These dialects form a layered pattern as seen in Figure 1.

Figure 1 shows that OWL Lite is a subset of OWL DL which is in turn a subset of OWL Full. OWL Lite is usually used to express simple classifications and relationships. OWL DL, or OWL Description Logic, contains all OWL constructs but has certain limitations necessary to guarantee computational completeness and decidability. OWL Full contains all OWL constructs.

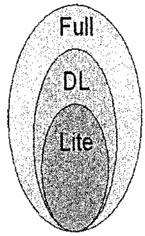


Figure 1. OWL Dialects Layering [5]

OWL Full cannot guarantee process completeness. The principal limitation of OWL DL is the restriction that classes cannot be instances. This is a necessary restriction for completeness.

An OWL ontology consists of a number of classes, properties and instances. Classes have definitions describing their characteristics. Properties have characteristics such as transitivity or functionality as well as some domain or range information. Individuals have a class membership, one or more relationships to other individuals and a concrete value.

3. Proposed Solution

3.1 Framework Architecture

Although a security solution such as this is almost completely composed of or related to access control, there are a number of other services which must be in place to offer a complete security framework. These services include a Key Management Service, an Encryption and Decryption Service and a Framework Management Service.

Figure 2 shows the proposed architecture of the proposed security framework which will be discussed in the remaining part of this section.

Key Management Service. The security framework will provide a service which will create, manage and store X.509 digital certificates. These certificates will be used as security tokens in requesting SOAP message headers to provide a non-repudiative user identity.

The Key Management Service will be designed and implemented using the XKMS (XML Key Management Specification) Standard from OASIS [6]. This provides two principle services:

- XML Key Information Service Specification (XKISS) — This service locates a public key in order to encrypt information for an individual or to verify signed information.

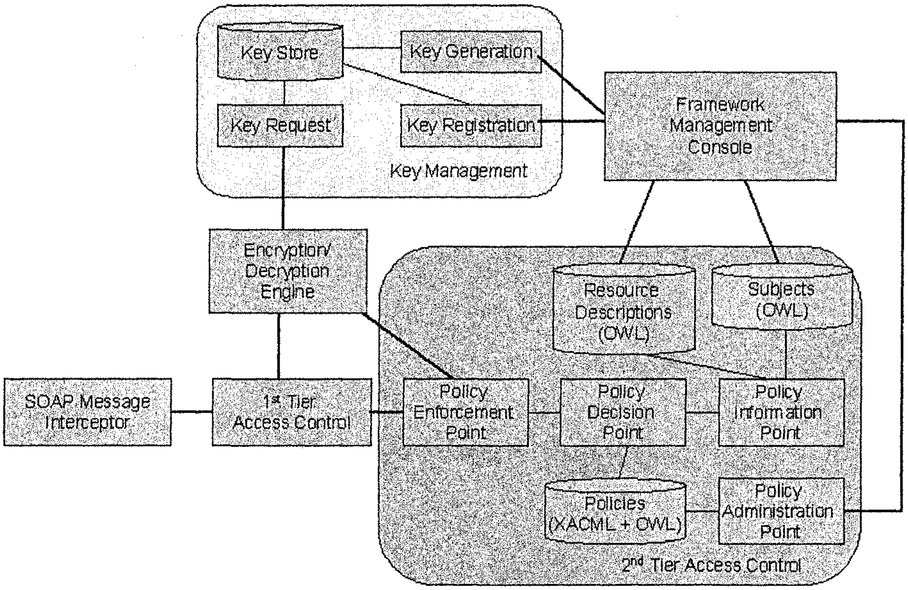


Figure 2. Proposed Security Framework Architecture

- XML Key Registration Service Specification (XKRSS) — This provides a number of services to register, recover, reissue and revoke keys.

As well as implementing the services specified in XKMS, keys will be stored locally by the security framework in order to reduce the interruption time between a user requesting a service and when that service is called. Keys for new users will be registered or created through the Framework Management Service.

Encryption and Decryption Service. As well as protecting information while in storage, the Security Framework must enforce a strict security policy on the confidentiality of information while 'on the wire'. All communication between remote clients and the Security Framework will be encrypted by the sender: be that the client or the framework. Additional encryption policies may be specified by the designers of the Web Service endpoints.

The Encryption and Decryption services will be exposed services from the encryption/decryption engine. This engine will encrypt and decrypt information according to a public key. If User_A makes a Web Service request to a Web Service managed by the Security Framework, the SOAP request is, at the very least, encrypted using our public key. The request will be subject to first tier authorisation (detailed in Section 3.2); upon successful authorization, the message will be fully decrypted by the encryption/decryption engine, using our private key, and passed to second tier authorization. The response returned to the requestor is encrypted using their public key which is located using the Key Management Service.

Framework Management Service. The Framework Management Service will be a HTTP and SOAP management centre used by the administrators of the Security Framework. It is essentially a front-end management of the different components of the framework. It will have five principal responsibilities:

- Uploading and registering or creating new keys
- Register valid Web Service endpoints
- Create/edit/remove access control policies
- Add/remove/edit users and semantic user descriptions
- Add/remove/edit semantic resource descriptions.

3.2 Access Control Model

The limiting of access to resources in this system will be done on a two tier level. The first tier

- Validates that the client requesting the resource is a registered user of the system and has a digital certificate to prove their identity
- Verifies that the requested Web Service end point exists and is registered with the security framework.

Failing either of these two checks will result in a security error and the request will not be passed to tier two.

The second tier of the access control model provides, rejects or limits access to the protected resources according to stored policies. This tier of the access control model will be similar in architectural design to that of the XACML standard. The purpose of the second tier is to define and control what the person can access; this can have three possible results:

- Access denied
- Full access granted
- Access granted but restricted.

The first two elements are straightforward. In the first case, the Web Service request is terminated and the response is returned to the user with the appropriate security exception; the second case results in the request being passed to the appropriate endpoint, free from all encryption, and the response is returned to the user that requested it. The third element is more complicated; although the request is permitted to be passed to the requested endpoint, the response must be examined according to the relevant policy rules to ensure illegal information access is not occurring. This scenario is common where a document of some sort is requested from an endpoint by a user that is allowed only limited access to the document. The Web Service request must be allowed to continue. However, the response must be intercepted to remove the appropriate fields. This is discussed in more detail in the following sections.

For this level of access control, it is necessary to provide a means of representing policies (access rules) which define the access rights that will be implemented in the second tier. We will continue to use the XACML standard, however augment it with semantics. Using a semantically-aware access control language increases the flexibility and power of the constructed policies. The next section defines the proposed policy language for use in this security framework; the following section presents how the subjects and resources, about which the policies are written, are described semantically. Policy evaluation is explained by describing how the policies written using the proposed language are evaluated according to the descriptions of the policy subjects and resources. The final section describes how access to documents can be limited without being fully removed.

Policy Language. The policy language for this security framework will be an OWL representation of XACML. The main expressions or constructs from XAMCL will be represented in OWL-DL atomic classes. From initial studies of both technologies the principal classes in our new language will be:

- **PolicySet** — This contains a set of policies; related policies will be grouped into sets.
- **PolicyCombiningAlgorithm** — When there is more than one policy in a policy set, there must be an algorithm to define precedence and conflict resolution.
- **Policy** — The policy contains a target, a set of rules and a rule-combining algorithm.
- **Target** — This specifies the subjects, resources, actions and environment to which the policy applies.
- **Subject** — This represents the subject to which the policy applies.
- **Resource** — This represents the resource to which the policy is protecting.
- **Action** — This is the resulting action which can or must take place when a policy is evaluated.
- **Environment** — This represents the environment attributes which must be present or absent from the request.
- **RuleCombiningAlgorithm** — This acts in the same way as the PolicyCombiningAlgorithm, on a rule level.

- **Rule** — Each rule contains a target, a condition or set of conditions, and an action or set of actions.
- **Condition** — This represents one condition of a rule.
- **Effect** — This represents the consequence of a rule evaluated to be true.
- **Obligation** — This represents an action which must take place as well as enforcing the access control decision. They can be defined at a policy or policy set level and are only executed if the appropriate policy or policy set is evaluated.

An OWL reasoning engine will be constructed to evaluate the semantically aware rules against the subjects and resources which are discussed in the next section.

Policy Subjects and Resources. The subjects and resource descriptions will be OWL ontologies that can be referenced in the policy rules. The subjects and resource descriptions will be domain specific. These ontologies will be built with an external tool, Protégé [7] for example, and will be uploaded to exposed interfaces using the management framework.

Policy Evaluation. Since the policy language and information representation is constructed using OWL-DL, Racer, an existing OWL reasoning engine, will be used as the basis for the Policy Decision Point. However, it will be necessary to extend this to provide for the obligations as described earlier.

Policy evaluation will take a number of steps. When a request is passed to the tier two access control, it is parsed by the Policy Enforcement Point (PEP) which will determine what the user is requesting. The PEP will then request a policy decision from the Policy Decision Point (PDP). The PDP will determine which policies apply to this request and source them from the policy store. To evaluate the policies, the PDP will have to request subject and resource description information from the Policy Information Point (PIP). Once the policies are evaluated, the decisions are returned to the PEP for enforcement.

Limited Document Access. The security offered by our security framework will be an XML element level. This fine grained level of access control is required in many of today's security environments, as there are many people at different levels of an organisation's hierarchy with different levels of access to documents, and even within one document. The documents will be defined semantically at an element level, which will enable the element level control to be decided at the evaluation by a semantic reasoner.

Enforcing this level of control is envisaged to occur in two steps, first at the point where the Web Service request enters the system, and the second as the response is leaving. Figure 3 shows the flow of what happens in this scenario. On the receipt of the request, first-tier access control is enforced; after passing this tier the second-tier of access control is enforced. This will have one of the three results defined in Section 3.2. In the case of the first two results (access denied or full access granted), the request will proceed as normal. In the case where there is a document or part of document requested from the Web Service, the service will be called normally but there will be a flag set in the response interceptor to prune the document being returned in the message, according to the semantic rules triggered in the reasoning engine.

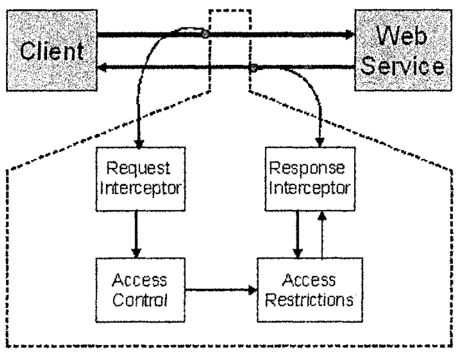


Figure 3. Limited Document Access Control Model

4. Case Study: Health Sector

Access control and privacy is critical in the health sector. There are innumerable documents and items of information in one active hospital. This fact, coupled by the countless levels of access by different groups, creates a monolithic task out of defining access control rights or rules. Enforcing these rules can be just as problematic. When the hospital administration eventually put a system in place, which is more often than not static, it is not portable to other hospitals in the health service. This is an activity that must be carried out at each location.

We propose the use of semantics in the management of access control of these systems. Standards already exist for representing information in the health sector; we will represent these standards in OWL and use them as our policy subjects and resources. The standard for information representation in the health sector is Health Level 7 (HL7) [8]. Figure 4 shows the core classes of the HL7 Reference Information Model (RIM). These six core classes will be the principle classes in the ontology, with numerous subclasses extending from them.

Bhavna Orgun [9] has developed a similar ontology using Protégé [7]. We will further this by representing HL7 in OWL.

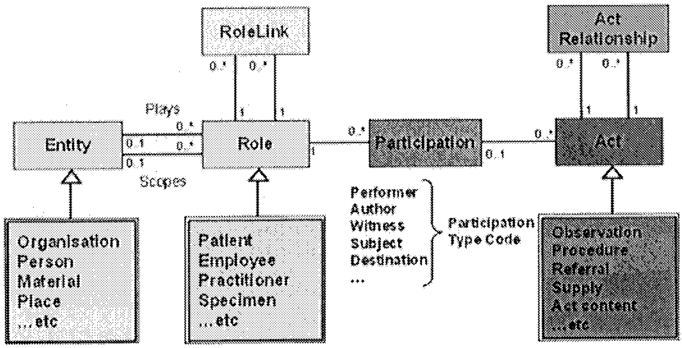


Figure 4. RIM Core Classes and Specialisations

The subjects and resources in our architecture are modelled as two separate items of storage. For the purpose of this case study, we will be representing all the information needed for policy reasoning in one ontology, as specified by HL7. Examples of rules which will need to be represented in our policy language are:

```
"Clinical information may only be accessed by clinical staff
"Nurses may only access information on a patient under their
care".
```

From a series of rules similar to this, our system will be able to determine, for example, if a particular nurse has access to the lab results for a particular patient.

5. Related Research

KAoS uses OWL for reasoning about policies. KAoS exploits ontologies for representing domain information describing organisations of people, agents and other computational actors. KAoS was initially designed as a policy language for complex software agents, but it is now being adapted to grid computing and Web Service environments. "Within the KAoS Policy Ontologies (KPO), a policy is an instance of the appropriate policy type (positive or negative authorization; positive or negative obligation) that defines the associated values for its properties" [10]. The KPO defines basic ontologies for actors, actions, groups, places and policies. KPOs are used for the analysis and inference of policies. The KAoS framework provides KPAT (KAoS Policy Administration Tool) for policy specification, modification and execution. While policies can be defined using KPAT, KAoS represents these policies in OWL. The user does not have to deal with policies at this level. KAoS can detect policy conflicts when they are being specified, using Stanfords Java Theorem

Prover (JTP) [11]. KAoS will try and resolve this conflict by placing an order on the policies.

Rei is a distributed policy language that enables every Web entity to specify policies for its access, for privacy, for entities it wants to communicate with, etc., which are enforced either by an internal policy engine or the policy engine on the platform on which it is running or with which it is registered [12]. Rei v2 is written using OWL-Lite. Rei however extends OWL-Lite to include logic-like variables. Policies defined in Rei are described in terms of [13]:

- permissions — able to do something
- prohibitions — not able to do something
- obligations — should do something
- dispensations — should not do something

These are grouped into permissibility and obligation. These four terms are known as deontic objects as defined in deontic logic. The Rei framework provides a policy engine that reasons about the policy specification. Upon loading, the Rei engine will detect any potential conflicts within policies.

Qin et al propose "an access control model for the Semantic Web that is capable of specifying authorizations over concepts defined in ontologies and enforcing them upon data instances annotated by the concepts" [14]. To this end, Qin et al propose a solution that not only grants access to a subject, on an object at an element, document and DTD level, but also at a concept level. The novel approach to access control by proposed by Qin er al is not limited to access restriction at a concept level; it also proposes the ability to propagate these access policies based on the semantic relationships among concepts or ontologies. They present an OWL-based access control language SACL (Semantic Access Control Language) as the language used to create authorisation policies in their proposed model. SACL is an extension of OWL. It has such additions as *SACL:higherLevelThan* and *SACL:lowerLevelThan*, to specify ordering, and *'canRead'* and *'readBy'* to specify privileges.

Parsia et al, in [15], propose a semantically-aware policy language by translating WS-Policy [16] into OWL-DL. They propose two translations; the first translates policies into OWL-DL classes. In the first case the WS-Policy grammar is encoded in OWL, whereas in the second case the actual formalism underlying the WS-Policy grammar is captured in OWL. To represent policies in OWL instances, Parsia et al define two particular OWL classes, one to represent WS-Policy assertions and the other to represent WS-Policy alternatives. Policy assertions usually deal with domain specific knowledge. Alternatives are groups of assertions, each of which must be satisfied by the requestor for the alternative to be satisfied. The second case "maps the WS-Policy fromalism directly in OWL" [15]. First of all policy assertions are mapped into OWL-DL atomic classes. Since assertions are now classes, relationships between these classes must be defined.

Damiani et al [17] outline how "current standard policy languages such as XACML can be extended" to be able to semantically define access control policies for the Semantic Web. They propose the use of RDF to make the XACML policies more semantically aware. They extend XACML to include data describing subjects and resources, to use RDF assertions or user defined properties and to define some policy processing information. Damiani et al's policy evaluation engine performs two principle activities; the comparison of the user assertions in the request and the user descriptions in the ontologies to identify appropriate policy rules; and the querying of resource descriptions to determine if the requesting user satisfies these rules.

6. Conclusions

Existing standards for access control are quite restrictive. Recent advancements in XACML have provided generic attributes of a requestor and resources but do not harness the expressive power and reasoning capabilities of the Semantic Web. We have presented a proposed access control model which we believe has selected the best attributes of other solutions in the area. We can see from the previous section that new ontologies in the Semantic Web, especially those written for the Web, are being built using one of the OWL languages. It is quite important to follow the trends of the community, in our opinion. We have therefore selected OWL as the language with which we will create our knowledge-bases.

The importance of standards cannot be emphasised strongly enough. Within the realm of Web Services security, XACML has become the front runner in defining access control. It is important for us to use this architecture, as agreed by W3C [18], as the basis for our proposed solution. By coupling XACML with OWL access control rules can be represented in a standard logic and can therefore benefit from the tools and expertise associated with popular standards such as these.

From the solutions described in Section 5 Rei, KAoS and the idea proposed by Parsia et al [15] are more concerned with policies that can be exchanged between communicating parties and how they can be enforced. This is certainly the nature of WS-Policy in the Web Services Security arena. We propose using XACML which is a standard for representing access control rules. Although exportable policies can be deduced from these rules, it is not the principal goal. Damiani et al [17] also base their solution on XACML yet represent the rules in RDF. OWL-DL yields maximum expressiveness without loosing computational completeness.

Acknowledgements. This work is funded by Enterprise Ireland as part of the Advanced Technology Research Program.

References

[1] Gray, Matthew. "Internet Growth Summary". Available at "http://www.w3.org".

[2] eXtensible Access Control Markup Language (XACML). Available at "http://docs.oasis-open.org/xacml/access_control-xacml-2_0-core-spec-cd-04.pdf".

[3] O'Neill M.*Web Services Security*, McGraw-Hill/Osborne, 2003.

[4] OWL Web Ontology Language. Available at "http://www.w3.org/TR/owl-features/".

[5] Klien, M. "Beyond RDF Schema". Available at "http://dit.unitn.it/ bouquet/kr-2005/slides-libro/wbkr_c6.pdf".

[6] XML Key Managemant Specification (XKMS). Available at "http://www.w3.org/TR/xkms/".

[7] Protégé. Available at "http://protege.stanford.edu/".

[8] Health Level 7 version 3. Available at "http://www.hl7.org/".

[9] Orgun, B. "http://www.ics.mq.edu.au/ borgun".

[10] Tonti, G., J. M. Bradshaw, R. Jeffers, R. Montanari, N. Suri and A. Uszok. "Semantic Web Languages for Policy Representation and Reasoning: A Conparison of KAoS, Rei, and Ponder". In Proceedings of *2nd International Semantic Web Conference*,Sanibel Islands, Florida, USA. 2003.

[11] JTP-Java Theorm Prover. Available at "http://www.ksl.stanford.edu/software/JTP/".

[12] Kagal, L., T. Finin and A. Joshi. "A Policy Based Approach to Security for the Semantic Web". In Proceedings of *2nd International Semantic Web Conference*, Sanibel Islands, Florida, USA. 2003.

[13] Kagal, L. "A Policy Based Approach to Governing Autonomous Behaviour in Distributed Environments". PhD Dissertation, University of Maryland, Baltimore, MD, USA. 2004.

[14] Qin, L. and V. Atluri. "Concept-Level Access Control for the Semantic Web". In proceedings of *ACM Workshop on XML Security*, Fairfax, VA, USA. 2003.

[15] Parsia, B., V. Kolozski and J. Hendler. "Expressing WS Policies in OWL". In proceedings of *14th International World Wide Web Conference*, Chiba, Japan. 2005.

[16] Web Services Policy Framework (WS-Policy). Available at "ftp://www6.software.ibm.com/software/developer/library/ws-policy.pdf".

[17] Damiani, E., S. De Capatani di Vimercati, C. Fugazza and P. Samarati. "Extending Policy Languages to the Semantic Web". In proceedings of *4th International Conference on Web Engineering*, Munich, Germany. 2004.

[18] World Wide Web Consortium (W3C), "http://www.w3.org".

SEMANTICALLY ENHANCED DISCOVERY OF HETEROGENEOUS SERVICES

A. Tsalgatidou, G. Athanasopoulos, M. Pantazoglou
University of Athens, Department of Informatics and Telecommunications

Abstract: Industrial application development approaches are striving for solutions that promote the rapid development of flexible and adaptable systems and the exploitation of legacy systems and resources. The Service-oriented Development (SOD) paradigm, a current trend in software development, could be beneficial to industrial application development approaches. However, the heterogeneity in existing standards and protocols for the discovery of the various service types is an obstacle for the use of SOD in industry. This paper addresses this issue by proposing a solution that supports the unified discovery of heterogeneous services and thus supporting the use of SOD in industry. The proposed solution comprises a generic service model (GeSMO), which facilitates the specification of heterogeneous services, a query language called Unified Service Query Language (USQL), based on GeSMO, which facilitates the unified discovery of heterogeneous services within heterogeneous service registries and a query engine called USQL Engine, that enables the execution of queries described in terms of the USQL, upon heterogeneous service registries.

Key words: Service-oriented Development, Heterogeneous Services, Web Services, P2P Services, Grid Services, Generic Service Model, Semantically-enhanced Service Discovery.

1. INTRODUCTION

Software engineering is gradually shifting to Service-Oriented Architecture (SOA) [SOA] and related technologies, in order to address critical contemporary issues imposed by the emergence of the Web, such as low cost application development and application interoperability. Industry's competitive environment needs technology solutions that will facilitate Rapid Application Development (RAD) and ensure application features such as flexibility and adaptability. Moreover, the exploitation and reuse of legacy systems constitutes a critical factor for the adoption and viability of these solutions. To satisfy the aforementioned requirements, industry seems to be embracing current trends of Service-Oriented Development, and it is expected that the emerging Semantic Web [SemWeb] will further accelerate the coalescence of the two worlds.

Nowadays, the Web bustles with services that are characterized by a high degree of diversity and heterogeneity. Web, Grid, and P2P services are continuously gaining momentum, yet, these are ruled by different and heterogeneous protocols and standards, making it difficult for them to interoperate. As a result, industry is intimidated in integrating and composing such diverse components for the utilization of service-oriented applications. Clearly, the full dynamics of these service technologies will be exposed and exploited by the industry, only when appropriate languages and tools emerge, which will render integration and interoperability among these technology areas feasible. Therewithal, services need to be discovered in order to be integrated in the context of an industrial application and, besides that, semantic annotations in service descriptions are required, in order to facilitate and automate the process of service discovery.

The provision of a framework that will encompass all previously mentioned requirements is expected to become the stepping stone to a new, service-oriented era in the world of industrial applications. SODIUM [SODIUM] forms an integrated solution for supporting and facilitating the comprehensive and unified visual composition, discovery, execution and monitoring of heterogeneous services. SODIUM platform comprises a set of languages as well as a set of individual, distributed and loosely-coupled components, which collaborate in order to support the aforementioned functionality.

In this paper, we focus on the service query language and its enacting search engine provided by SODIUM, which, combined, enable the unified and semantically enhanced discovery of diverse types of services over heterogeneous registries and/or networks. The results of such discovery can then be used for the development of service compositions in industrial environments.

The rest of this paper is structured as follows: A *motivating scenario* is presented to demonstrate how a real industrial application can be developed according to a Service-Oriented Architecture, by utilizing various heterogeneous services, as well as how it benefits from such an approach. Next, a *Generic Service Model* is described, offering a common point of reference for the various types of services. Following that, we introduce a

Unified Service Query Language catering for the discovery of heterogeneous services that are compliant with the model previously discussed, as well as its *enacting engine*. Based on the motivating scenario, examples on using the language and the engine are provided, in order to showcase the various assets of our framework. Consequently, we include a brief section with related work in service definition and discovery and finally the paper is closed with our conclusion statements.

2. MOTIVATING SCENARIO

An appropriate domain for the application of service-oriented computing is the automobile industry. Car manufacturers and their suppliers face many significant challenges, including pressure to reduce cost and time to delivery in the supply chain. The dynamic nature of such supply chains and the heterogeneity among the systems of the respective stakeholders are some of the obstacles that a system developer has to face.

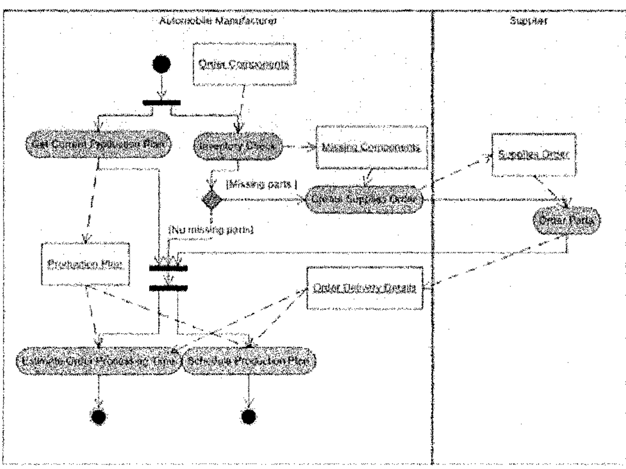


Figure 1. Order processing flow example

A crucial task that is usually met in an order processing workflow is the estimation of the processing time for a given order. In a car manufacturing industry there could be a plethora of requests for such calculations, which have to be answered promptly. However, the calculation of the order processing time is a computation intensive task that is depending on many factors such as the existence of all necessary components, the delivery time of non-existing components, the current factory production plans, order priority, etc.

A simplified workflow, which calculates the processing time of an order, is presented in *Figure 1.* According to this scenario, the workflow takes as input a list with the order's components that need to be provided. The workflow begins with the execution of two parallel tasks; one task provides the current production plans of the factory and the other checks whether the factory's warehouses have all the necessary materials. If some necessary materials are missing, the workflow continues with the preparation of an order for supplies and the submission of that order to a supplier which returns the necessary order details such as cost, delivery time and shipment method. Upon the completion of the aforementioned tasks, the workflow goes on with the execution of two parallel tasks, which estimate the order processing time and reschedule the production plan respectively. The outputs of the workflow are the estimation of the order completion time and the reformed production plan which takes into account the new order.

The tasks modeled in this scenario do not have to be developed from scratch. They could be performed by already existing services that may be available over the Internet. For example, the functionality required by the tasks *"Get Current Production Plan", "Create Supplies Order"* and *"Order Parts"* may be provided by respective web services which are registered in a web service registry (e.g. UDDI) and are offered by various providers. . The *"Inventory Check"* task could be performed by a P2P service that is provided by a P2P network that exists between the manufacturer's warehouses. Last but not least, the functionality required by the *"Estimate Order Processing Time"* and *"Schedule Production Plan"* tasks could be provided by grid services which are utilizing the resources of a grid network where the car manufacturing organization is participating.

In order to be integrated in the aforementioned workflow, services have to be firstly discovered. However, service discovery is not an easy task, due to the heterogeneity and incompatibility between the existing description and discovery protocols and standards for web services, grid services and P2P services. In the following, we describe our solution to this problem that comprises a Generic Service Model, a Unified Query Language and a respective enacting engine.

3. GENERIC SERVICE MODEL

Although, service-oriented technologies (e.g. Web, Grid and P2P services) comply with the same paradigm, they adhere to different models, having different characteristics and different properties [GeSMO][OGSI2WSRF][P2P&Grid]. Moreover, their heterogeneity spawns across other aspects such as architecture, supported protocols and standards, infrastructure, semantics and quality of service (QoS). This diversity makes the integration of different services a strenuous task.

Therefore, in order to remove this burden from a system developer a generic service model (GeSMO) incorporating features and properties of all service-oriented technologies needs to be provided. This model will facilitate the specification of any type of service and the mapping and/or association of service features of one technology to the other.

3.1 Service Model Structure

An assessment of the service models of the addressed service-oriented technologies brings up a set of common features and properties that may be regarded as the common denominator of the web, grid and P2P services, which are the service types being addressed in this paper. Nevertheless, apart from this set of common features there are a lot of discrepancies among the various types of services.

Thus, a layered structure seems to be appropriate for the specification of GeSMO comprising a core layer with common features of all service-oriented technologies and with appropriate extensions providing for the specific features and properties of each of the addressed service types. *Figure 2* illustrates the structure of GeSMO. Furthermore, crosscutting issues such as Semantics, Quality of Service, Trust, Security and Management are pertinent to all types of services and may be related to any element of the service model.

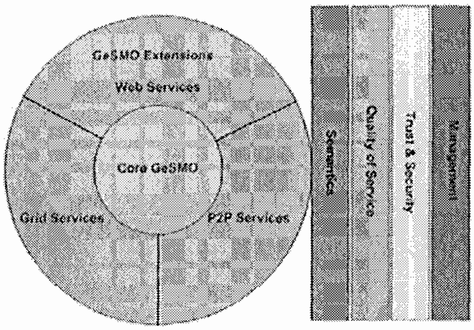


Figure 2. Generic Service Model Structure

In the following we present each of the identified layers and the interrelationships among their elements.

3.2 Core Service Model Concepts

After a thorough investigation of the current state of the art in service technology, we came up with a set of features that seem to be pertinent to all types of services. As it is illustrated in *Figure 3* a service is regarded as a software system that exchanges messages, which are usually XML-formatted, it resides at a specific network address and it has a description that may be an XML- formatted document.

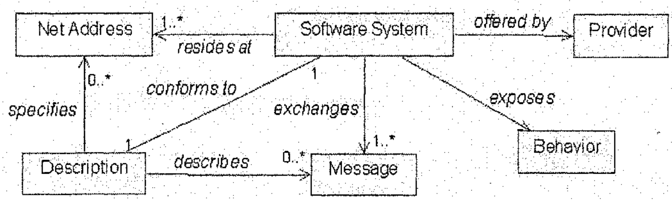


Figure 3. Service model

Service descriptions, which may be semantically and/or quality of service enhanced are published in service registries which are used by requestors for the discovery of appropriate services. A service description contains information that can be used for the identification and invocation of a service (*Figure 4*).

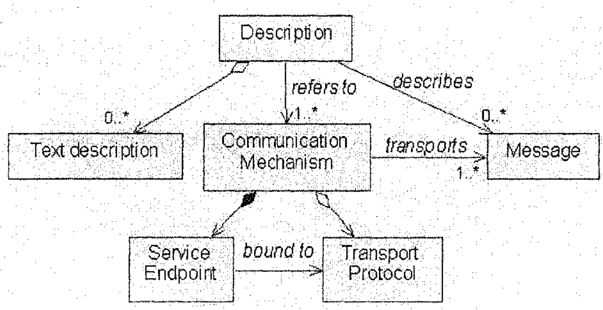


Figure 4. Service description structure

A service description conveys information, such as the specific endpoint that a service resides, the protocol that can be used for the message exchange and text descriptions providing human readable information about the service. In some cases, the specification of the message exchange mechanism may not be explicitly described, e.g. in P2P services an implied scheme is used. In these cases information related to the service endpoint or the protocol used is inferred by the underlying platform.

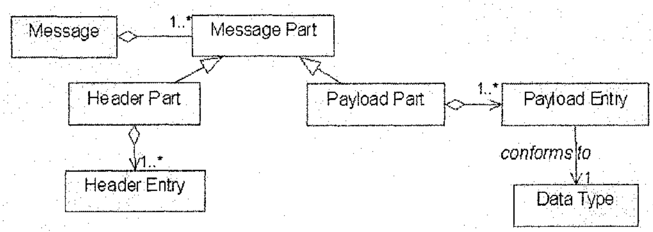


Figure 5. Message Structure

Exchanged messages are composed of two parts: header and payload information (*Figure 5*). The header part normally conveys information that is manipulated by the intermediate nodes/middleware transporting the messages. Such information may be routing information, security or transaction context information, etc. The payload part of a message conveys information that is consumed by the service or its client. This information is application specific and it normally abides by data types that are specified by the platform (e.g. Strings, Integers, etc) or the service provider (e.g. Addresses, Contacts, etc).

Service description documents contain additional information that facilitates the invocation of services. Services implement specific interfaces which describe the operations that are offered by a service (see *Figure 6*). These operations exchange messages, which convey information that abides by specific data types, with the service clients. These messages could be either incoming or outgoing with respect to the service. This information is also included in a service description document as it is necessary for the invocation of a service.

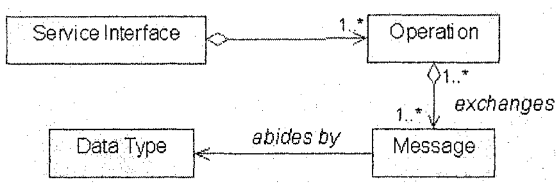


Figure 6. Service description elements

We have to note here that, service invocation information is not provided by all service type descriptions, e.g. by P2P service descriptions, as it can be either inferred by the underlying infrastructure or by the service implementation.

4. UNIFIED SERVICE QUERY LANGUAGE (USQL)

The *Unified Service Query Language (USQL)* is an XML-based [XML] language enabling requestors to formulate queries asking for available services. The language specification describes both requests and corresponding responses. The main contribution of USQL lies in that it follows a unified approach to expressing queries as regards the heterogeneous types of services. This is achieved with the language abiding by the core concepts introduced by GeSMO, as far as the abstract definition of a service is concerned. On the other hand, USQL responses may be easily extended so as to provide the concrete information for invoking the service, with respect to its type. Thanks to its flexible and extensible design, USQL can consolidate virtually any GeSMO-compliant type of service, thus providing service-oriented industrial applications with a wide lookup range regarding candidate services that could be integrated and used for fulfilling a specific task.

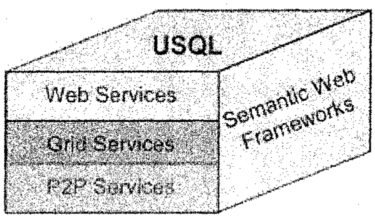


Figure 7. Orthogonal position of USQL with respect to services and semantic frameworks

USQL currently addresses - but is not limited to - Web, Grid, and P2P services, aiming at applying semantically enhanced queries for discovering them. As depicted in *Figure 7*, USQL is orthogonal with respect to these diverse service types and their description protocols; moreover, semantic concepts supported by the language are generic enough so as to map to most well known emerging semantic frameworks, such as OWL-S [OWL-S] and WSMO [WSMO], thus enabling the exploitation of their capabilities.

4.1 Semantics in USQL

Although syntactic information suffices for the invocation of a service, confining a service query to syntactic matching yields in most cases to scrappy results; the response to a query based on syntactic information either misses services, or contains services which are actually irrelevant to the initial request [ESSW03]. Furthermore, the limited expressiveness of syntactic information is an obstacle when applying service discovery at runtime. To tackle such cut-backs, USQL enhances service requests with semantic information, in order to provide users with more expressive means. The supported semantics consist of domain-specific annotations which are bound to service operations and their respective input/output. In addition, USQL provides a set of elements and structures to allow for the application of QoS requirements in the search criteria, in order to refine service discovery and selection.

Briefly, USQL provides the following features for semantically annotating service requests:

- ***Domain*** – implemented by an element called *ServiceDomain*, this feature enables requestors to specify an application domain for the requested services and thus to semantically enhance the query and to confine the search range. This is the first step towards overcoming scrappy and irrelevant results.
- ***Input/Output*** – the *Input/Output* elements enable requestors to apply semantic criteria regarding the expected input/output of an operation offered by a service.
- ***Capability*** – the *Capability* element enables requestors to apply semantic criteria regarding the expected capability (i.e. the abstract functionality) of an operation offered by a service.

USQL introduces a set of operators that can be applied to semantic elements during service discovery, determining the type of inference rules that should be employed for reasoning purposes. More specifically, the following types of inference are supported by the language:

- ***exact*** – indicates that the element's value must be an exact match of the value of the corresponding element in the service advertisement.
- ***abstraction*** – indicates that the element's value must be subsumed by that of the corresponding element in a service advertisement.
- ***extension*** – indicates that the element's value must subsume that of the corresponding element in a service advertisement, besides exact matching.

USQL defines a generic type for all supported semantic elements, which contains the following attributes:

- ***typeOfMatch*** – applies any combination of the aforementioned operators to the semantic element, indicating the type of inference that must be employed during the discovery process, in order to determine if a service satisfies the specific semantic requirement.
- ***nullAccepted*** – specifies whether services not including the corresponding element in their description should be further processed, and potentially included in the results, or not.
- ***ontologyURI*** – associates the value of the semantic element with an existing ontology, identified by a URI.

By employing these relatively simple artefacts, USQL enriches service queries semantically, allowing requestors to express their requirements in a more explicit way, thus yielding to concrete and consistent results. Nevertheless, by keeping semantics support to this level of simplicity, the language retains its openness and orthogonal position as regards existing and emerging types of services and semantic frameworks. In the following paragraph, we demonstrate how USQL can be used to formulate semantically enhanced queries looking for appropriate services that would satisfy the requirements imposed by the previously presented motivating scenario.

4.2 Using USQL

As shown in the motivating scenario, the first step in order to calculate the processing time for an order in the domain of automobile is to retrieve the current production plan. Thus, a Web service with the specific output is needed to fulfil the task. The following USQL request (*Figure 8*) encompasses these details with the use of semantic annotations, in order to find the most appropriate service for the job:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<USQL version="1.0">
  <find_servicesRequest>
    <Where>
      <Service serviceType="WebService">
        <ServiceDomain ontologyURI="http://http://nkua/sodium/usql/engine/ontology/dco"
          typeOfMatch="exact extension">
          AutoMobile</ServiceDomain>
        <Operation>
          <Capability ontologyURI="http://nkua/sodium/usql/engine/ontology/dco"
            typeOfMatch="exact">
            GetCurrentProductionPlan</Capability>
          <Output>
            <semantics ontologyURI="http://nkua/sodium/usql/engine/ontology/dco"
              typeOfMatch="extension">
              ProductionPlan</semantics>
          </Output>
        </Operation>
      </Service>
    </Where>
  </find_servicesRequest>
</USQL>
```

Figure 8. Example USQL request for Web services

```xml
<?xml version="1.0" encoding="UTF-8"?>
<USQL version="1.0">
  <find_servicesRequest>
    <Where>
      <Service serviceType="P2PService">
        <ServiceDomain ontologyURI="http://http://nkua/sodium/usql/engine/ontology/dco"
          typeOfMatch="exact extension">
          AutoMobile</ServiceDomain>
        <Operation>
          <Capability ontologyURI="http://nkua/sodium/usql/engine/ontology/dco"
            typeOfMatch="exact">
            InventoryCheck</Capability>
          <Input>
            <semantics ontologyURI="http://nkua/sodium/usql/engine/ontology/dco"
              typeOfMatch="exact extension">
              ListOfComponents</semantics></Input>
          <Output>
            <semantics ontologyURI="http://nkua/sodium/usql/engine/ontology/dco"
              typeOfMatch="exact extension">
              MissingComponents</semantics></Output>
        </Operation>
      </Service>
    </Where>
  </find_servicesRequest>
</USQL>
```

Figure 9. Example USQL request for P2P services

Given a set of required components for the production of a car, the workflow needs to access the established P2P network and look for a service that will enable checking against warehouses for missing components. The

USQL message expressing a request for such a service is depicted in *Figure 9.*

The estimation of the time that is required for processing an order is a demanding operation in terms of processing power, due to its complex calculations. Hence, a Grid service would be the perfect candidate for carrying out this task. *Figure 10* depicts the respective USQL request.

```
<?xml version="1.0" encoding="UTF-8"?>
<USQL version="1.0">
  <find_servicesRequest>
    <Where>
      <Service serviceType="GridService">
        <ServiceDomain ontologyURI="http://http://nkua/sodium/usql/engine/ontology/dco"
          typeOfMatch="exact extension">
          AutoMobile</ServiceDomain>
        <Operation>
          <Capability ontologyURI="http://nkua/sodium/usql/engine/ontology/dco"
            typeOfMatch="exact">
            EstimateOrderProcessingTime</Capability>
          <Input>
            <semantics ontologyURI="http://nkua/sodium/usql/engine/ontology/dco"
              typeOfMatch="exact extension">
              ProductionPlan</semantics></Input>
          <Input>
            <semantics ontologyURI="http://nkua/sodium/usql/engine/ontology/dco"
              typeOfMatch="exact extension">
              OrderDeliveryDetails</semantics></Input>
          <Output>
            <semantics ontologyURI="http://nkua/sodium/usql/engine/ontology/dco"
              typeOfMatch="exact">
              OrderProcessingTime</semantics></Output>
        </Operation>
      </Service>
    </Where>
  </find_servicesRequest>
</USQL>
```

Figure 10. Example USQL request for Grid services

As shown in the above USQL request examples (Figures 8-10), the values of all semantic USQL elements are typically references to classes or instances within a specific ontology. The value of the *ontologyURI* attribute indicates the ontology, while the actual value of the XML element is the ID of a specific class or instance within that ontology. Thus, the combination of these values constitutes the *Uniform Resource Identifier (URI)* of the concept that is being used for the population of a semantic USQL element, taking the form of "*ontologyURI#elementValue*".

5. USQL ENGINE

The *USQL Engine* is a service search engine, based on the USQL language, which provides the means for accessing and querying heterogeneous service registries and/or networks in a unified and standards-based manner. The functionality offered by the engine is exposed as a Web service; thus, abiding by the SOA principles, the USQL Engine itself may be integrated in the context of a service-oriented industrial application allowing for automated service discovery.

The main concept underlying the USQL Engine framework is the abstraction regarding registry details, from the requestor's perspective. This is achieved with the adoption of a domain-centric categorization of the various supported registries, depending on the service advertisements they host. Domain information provided by the requestor is exploited by the engine so as to identify, access and query the appropriate registries in a transparent manner.

The USQL Engine follows an architecture distinguished by its high degree of openness and extensibility, which is achieved by applying plug-in mechanisms in order to accommodate virtually any type of service, registry, as well as their governing protocols and standards. The plug-ins used for this purpose can be integrated in a flexible manner, so as to enable different configurations and to broaden the range of supported registries.

Many of the tasks accomplished by the engine during service discovery are facilitated by an *Upper Ontology*, which forms a constituent part of the overall framework and reflects the domain-driven aspect of our approach. The ontology classes and properties mirror the semantic concepts supported by USQL and thus, the ontology is directly used for the population of semantic elements within USQL requests. Upon submission of a USQL request to the engine, the implicit identification of the registries and/or networks where the query will be forwarded is carried out by navigating in the ontology, making use of the domains specified by the requestor, and finding the registries that have been registered therein as belonging to these domains. Finally, reasoning during the matchmaking process is performed based on the structure and rules imposed by the upper ontology. *Figure 11* depicts the structure of the *USQL Engine Upper Ontology*:

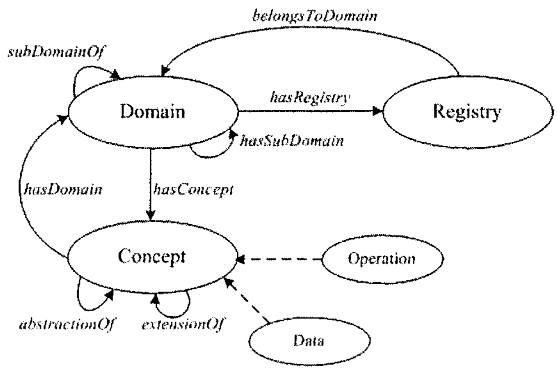


Figure 11. The USQL Engine Upper Ontology

The upper ontology consists of the following classes:

- ***Domain***: represents the domain where a service belongs to.
- ***Registry***: represents a registry/repository/network holding service advertisements.
- ***Concept***: represents Domain-specific concepts that may be used for describing services. A concept may be either an *Operation* or a *Data* description, related to a specific domain. Therefore, two subclasses of the *Concept* class are defined:
    - ***Operation***: represents an abstract functionality that is specific to a domain.
    - ***Data***: represents a piece of information that is specific to a domain.

It is worth noting that the *Concept* class is never instantiated. Instead, it serves as an abstraction to hold properties that are common to both operations and data.

The *Domain* class has the following properties:

- ***hasRegistry***: Takes as value a *Registry* instance. A domain may have zero or more associated registries.
- ***hasConcept***: Takes as value either a *Data* or an *Operation* instance.
- ***subDomainOf***: Takes as value a *Domain* instance. A domain may be the sub-domain of at most one parent domain.
- ***hasSubDomain***: Takes as value a *Domain* instance. A domain may have zero or more sub-domains.

Hence, with the use of the *subDomainOf* and *hasSubDomain* properties we can build a bi-directional tree, i.e. a domain hierarchy, which is easy to navigate.

The *Registry* class has the following property:

- ***belongsToDomain***: Takes as value a domain instance. A registry may belong to one or more domains, depending on the kind of service advertisements it holds.

The *Concept* class has the following properties:

- ***hasDomain***: Takes as value a *Domain* instance. A concept must belong to at least one domain.
- ***abstractionOf***: Takes as value either a *Data* or an *Operation* instance. A concept may be the abstraction of zero or more others concepts. More specifically:
  Concept A is an abstraction of concept B, if A subsumes B
- ***extensionOf***: Takes as value either a *Data* or an *Operation* instance concept. A concept may be an extension of at most one other concept. More specifically:
  Concept A is an extension of concept B, if A is subsumed by B.

The *abstractionOf* and *extensionOf* properties allow for the construction of data and operation concept hierarchies. Hence, the upper ontology provides a tree structure for both domains and their concepts, which is useful when applying reasoning and inference during service discovery.

6. RELATED WORK

Related work with respect to this paper can be classified into work related to the provision of a service model and work related to service discovery.

As far as work related to service models is concerned, one major approach is that of W3C. W3C's Architecture Working group in [w3c2004] has established a model for the specification of web services. The core concepts of the generic service model presented in this paper have a lot of similarities with the concepts of the W3C model. However, our model remains abstract enough allowing thus for extensions that are able to support P2P and Grid services, whereas the W3C's model is confined to web services and lately to grid services complying with the WSRF specification [WSRF].

OASIS has recently announced the formation of a task group working on the specification of a service-oriented architecture reference model. This group has produced a working draft version of the reference model [SOARfMo]. However, the provided document is in draft version and no useful results can come out of it.

Service discovery on the other hand, is currently performed in the areas of Web, Grid, and P2P services with the use of custom APIs and discovery mechanisms offered by registries and networks.

UDDI [UDDI] has become the registry model of choice for publishing and discovering Web services. The framework provides for keyword-based search, allowing requestors to look for services according to their provider, classification, name, description etc. UDDI does not take into consideration semantic, as well as QoS descriptions and properties of services, although it provides a structure allowing the incorporation of arbitrary service descriptions within the registry. To exploit this feature, many efforts have been made towards integrating semantic annotations in UDDI; Paolucci et al. have proposed a way to map the OWL-S profile and process model in UDDI [Paolucci].

JXTA [JXTA] comprises a set of open, generic and implementation-independent P2P protocols allowing any device to communicate and collaborate as a peer over a network. One of the most important contributions of the JXTA framework is the explicit definition of P2P services, with the use of XML-based JXTA advertisements. This enhancement allows for the application of service discovery within JXTA networks, with the use of the standard discovery service provided by the platform. Still, JXTA protocols and advertisements are generic and very limited with respect to syntactic information, and moreover they do not detail crucial aspects of a P2P service like semantics and QoS, which could be exploited during service discovery.

JAXR [JAXR] provides a uniform and standard API for accessing different kinds of XML registries. On the other hand, the evolution of frameworks such as OWL-S and WSMO enables formulation of semantically-enhanced service requirements that can be checked against service offerings also described with the use of these frameworks.

Currently, a number of search engines have been proposed and/or implemented, all of which are activated in the area of Web services, without taking into account other existing types of services [Woogle], [SalCentral], [BindingPoint].

Woogle, a search engine for Web services, enables similarity search by employing a set of matching and clustering algorithms with promising experimental measures and results. However, Woogle does not cater for the discovery of other types of services, while, in the context of Web services, matchmaking relies on the information provided in the WSDL [WSDL] file and the UDDI entry only, without taking into account and exploiting semantics.

Like Woogle, other existing Web service search engines also focus on UDDI and WSDL descriptions of Web services, thus confining their queries to syntactic-based matchmaking only. SalCentral, a WSDL aggregator and analysis engine, allows for WSDL and XSD [XSD] based service lookups, while BindingPoint categorizes and provides access to a large number of Web services.

Nevertheless, it is clear that service-oriented development lacks a query language that would enable accessing and querying heterogeneous registries in a unified, standards-based manner. Moreover, exploitation of semantics and QoS within service descriptions proves to be a crucial part of service discovery. USQL and its enacting engine address these issues and constitute a stepping stone to the unification of the various heterogeneous service areas.

7. SUMMARY AND CONCLUSIONS

Industrial applications impose many requirements that can be met by following the SOA paradigm. Moreover, as shown in the scenario presented in this paper, a service-oriented industrial application will most probably consist of various heterogeneous services, which in turn may be described with the use of different semantic frameworks. Currently, most of the emerging semantic frameworks apply to the Web Service paradigm, without supporting directly other types of services. Yet, discovery of P2P as well as Grid services could be greatly facilitated by the accommodation of semantics, as it has been argued in this paper. This heterogeneity in existing service-oriented frameworks, protocols and standards, particularly in the area of service discovery constitutes a major obstacle towards the use of SOA paradigm in the development of industrial applications.

The solution presented in this paper, comprising a generic service model (GeSMO), a compliant query language (USQL) and its supporting engine provides for a unified way of discovering such diverse kinds of services, facilitating their interoperability and enabling their integration in industrial environments. More specifically, GeSMO facilitates the specification of heterogeneous services, while the USQL and its supporting engine, although at their early stages, aspire to enable the unified service discovery over heterogeneous registries and/or networks. The language inherits from the service model features such as abstraction, generality, openness, and extensibility, so as to allow for the seamless unification of the various types of services with respect to discovery. Moreover, we showed how the application of a generic set of domain-centric semantics can enhance service requests and flavor the task of service discovery with transparency, regarding the nature of the registries and networks that are being looked up.

8. ACKNOWLEDGEMENT

This work is partially supported by the European Commission under contract IST-FP6-004559 [SODIUM].

9. REFERENCES

[w3c2004] D. Booth, et al. Web Services Architecture, W3C Working Group Note, Feb 2004, http://www.w3c.org/ws-arch/

[AlCa2004] G. Alonso, et al, Web Service: Concepts, Architectures and Applications, Springer-Verlag, 2004

[SOA] SOA, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm

[SemWeb] T. Berners-Lee et al, *"The Semantic Web"*, Scientific American, May 2001.

[SODIUM] SODIUM http://www.atc.gr/sodium

[XML] XML, Extensible Markup Language, http://www.w3.org/XML/

[OWL-S] OWL-S http://www.w3.org/Submission/OWL-S/

[WSMO] WSMO, Web Service Modeling Ontology, http://www.wsmo.org/

[WSRF] K. Czajkowski, et al, The WS-Resource Framework, ver 1.0, http://www.globus.org/wsrf/specs/ws-wsrf.pdf

[SOARfMo] C. M. MacKenzie, et al Service-oriented Architecture Reference Model, working draft 07, OASIS, May 2005

[UDDI] UDDI, Universal Description, Discovery and Integration, http://www.uddi.org

[Paolucci] Massimo Paolucci, et al, "Importing the Semantic Web in UDDI", Proceedings of Web Services, E-business and Semantic Web Workshop, 2002

[JXTA] JXTA, Juxtapose Technology, http://www.jxta.org

[JAXR] JAXR, Java API for XML Registries, http://java.sun.com/xml/jaxr/index.jsp

[Woogle] Xin Dong et al, *Similarity Search for Web Services*, Procs of VLDB 2004, Canada

[WSDL] WSDL 1.1, W3C Note 15 March 2001, http://www.w3.org/TR/wsdl

[SalCentral] Salcentral, http://www.salcentral.com

[XSD] XSD, XML Schema Definition, http://www.w3.org/XML/Schema

[BindingPoint] Binding point, http://www.bindingpoint.com/

[ESSW03] T. Pilioura, A. Tsalgatidou, A. Batsakis, *"Using WSDL/UDDI and DAML-S in web service discovery"*, Proceedings of the WWW 2003 Workshop on E-Services and the Semantic Web, Budapest, Hungary,May 20, 2003.

[GeSMO] A. Tsalgatidou, G. Athanasopoulos, M.Pantazoglou, et. al D4: Generic Service Model Specification, SODIUM (IST-FP6-004559) Project's Deliverable, Jun 2005

[P2P&Grid] I. Foster, A. Lamnitchi, On Death, Taxes and the Convergence of Peer-to-Peer and Grid Computing, Presented in 2nd International Workshop on Peer-to-Peer Systems (IPTPS03), Berkley, USA

[OGSI2WSRF] K. Czajkofwski, I. Foster, et. al. From Open Grid Services Infrastructure to WS-Resource Framework: Rafactoring and Evolution, March 2004

WEB SERVICE COMPOSITION ALGORITHM BASED ON FIX-POINT THEOREM

YanPing Yang, QinPing Tan, Feng Liu, JinShan Yu
Computer College of National University of Defense Technology
Changsha, Hunan, P.R.China
yanpingyang@nudt.edu.cn

Abstract: A number of web services are now available and it therefore seems natural to reuse existing web services to create composite web services. The key to the problem of web services composition is how to model the input and output data dependency of candidate web services and how to satisfy that of a service request by composition efficiently. In this paper we propose an algorithm based on the concept of invocation layer and Knaster-Tarski fixpoints theorem, which can be used to get the least invocation layers of candidate web services to satisfy the given service request. Then we design another search algorithm based on A* procedure to find the best composition ways according to the invocation layers.

Key words: web services, composition, algorithm, fixpoints, A*

1. INTRODUCTION

A web service is a software system designed to support interoperable machine-to-machine interaction over a network. There might be frequently the case that a web service does not provide a requested service on its own, but delegates parts of the execution to other web services and receives the results from them to perform the whole service. In this case, the involved web services together can be considered as a composite web service.

All-sided development process for composite web services involves solutions to several problems, which, generally speaking, are discovery of useful candidate web services, calculation of their possible composition, and execution of the new generated Web Service. The work presented in this paper is providing concrete approaches in calculation of web service composition. Much research work are devoted to this regard, and researchers use Petri Nets [8, 9], linear logic [10], state charts [1] or finite state machines [2] to model and execute composite Web Service. The large number of works in this area confirms the emerging interest in web services as service-oriented software artifacts, and their composition.

In this paper, we are interested in studying how web services can be composed to provide more complicated features. We propose an algorithm based on the concept of invocation layer and fixpoints theor- em, which can be used to get the least invocation layers of candidate web services to satisfy the given service request. Next, we design another search algorithm based on A* procedure to find the best composition ways according to the invocation layers. Meanwhile, we analyze the implementation issues of the algorithms.

The remainder of this paper is organized as follows: Section 2 introduces our motivation and Section 3 describes our algorithm generating the least invocation layers and analyzes the correctness and performance of it. Section 4 describes our search algorithm base on A* procedure and Section 5 proposes to use Bloom Filter to implement the operations of sets in the algorithms. Finally, conclusions and future plans are given in Section 6.

2. MOTIVATION

Web services are described in the Web Services Definition Language (WSDL) [5].Considering the following WSDL fragment of a Web Service:

```
<message name="findCloseRestaurant_Request">
  <part name="custAddress" type="xs:string"/>
  <part name="foodPref" type="xs:string"/>
</message>
<message name="findCloseRestaurant_Response">
  <part name="restaurantName" type="xs:string"/>
  <part name="restaurantAddress" type="xs:string"/>
  <part name="restaturantPhone" type="xs:string"/>
</message>
<portType name=" findCloseRestaurantPortType">
 <operation name="findCloseRestaurant">
  <input message="findCloseRestaurant_Request"/>
  <output message="findCloseRestaurant_Response"/>
 </operation>
</portType>
```

From this fragment, we can find that there are three pieces of semantic information which are vital to a web service: its input parameter set, output parameter set and data dependency information between different inputs and outputs. In practical, we can store a web service having several inputs and outputs dependency relationships as different items in repository, which will not bring any effect except for simplicity. So each web service ws can be defined formally as follows:

Definition 2.1 **(Semantic Web Services)** *A semantic web service is named as ws. Let ws_{in} with $ws_{in} = \{I_1, I_2, ..., I_p\}$ be its set of input parameters. Let ws_{out} with $ws_{out} = \{O_1, O_2, ..., O_q\}$ be its set of output parameters. Then ws can be denoted formally as:*

$$ws = < ws_{in}, ws_{out} >$$

Likewise, we can define formally a composition request r as follows.

Definition 2.2 **(Semantic Service Request)** *A service request is named as r. Let r_{in} with $r_{in} = \{A_1, ..., A_m\}$ be its set of available or existing input parameters. Let r_{out} with $r_{out} = \{D_1, ... D_n\}$ be its set of desired output parameters. Then r can be denoted formally as:*

$$r = < r_{in}, r_{out} >$$

Definition 2.3 **(Functions getIn and getOut)** *Let WS be the set of all available web services which can be found from a local file system, resources referenced by URIs or provided by a repository such as UDDI. The functions are mapped from a web services to the set of input parameters or output parameters respectively. They are denoted formally as follows:*

$$getIn: WS \rightarrow \{ws_{in} | ws \in WS\}, \quad getOut: WS \rightarrow \{ws_{out} | ws \in WS\}$$

In this paper, we look a service request as a special kind of web service, so function getIn and getOut can also act on service requests and get the available or desired parameters set respectively.

If we can discovery a web service ws satisfying a given service request r, then ws must be invoked using the existing parameters of r and produce the desired parameters of r. We define the conditions under which a web service ws satisfies a given semantic service request r as a predication *FullySatisfy*:

Definition 2.4 **(Predication FullySatisfy)** *Let WS be as Definition 2.3 and RQ be all service requests. $ws \in WS$ and $r \in RQ$. FullySatisfy is a predicate $FullySatisfy: WS \times RQ \rightarrow Bool$ having the following definition:*

$$FullySatisfy\ (ws, r) = true\ iff\ (getIn(ws) \subseteq getIn(r)) \wedge (getOut(ws) \supseteq getOut(r))$$

In practice, however, it is often impossible that one web service can fully satisfy the given request. Then, one has to combine multiple web services that only partially satisfy the request. Given a request *r* and two web services *x* and *y*, for instance, suppose one can invoke *x* using inputs in *getIn(r)*, but the output of x does not have what we look for in *getOut(r)*. Symmetrically, the output of *y* generates what we look for in *getOut(r)*, but one cannot invoke y directly since it expects inputs not in *getIn(r)*. Furthermore, using initial inputs of *getIn(r)* and the outputs of *x*, one can invoke *y* (i.e., $(getIn(r) \cup getOut(s)) \supseteq getIn(y)$).So the request *r* can be satisfied by the invocation layers of: $r \rightarrow \{x\} \rightarrow \{y\}$ .We define the conditions above as a predication *LayeredlySatisfy*.

Definition 2.5 (Predication LayeredlySatisfy) *Let r be as definition 2.4.* $S_1,\ldots,S_n$ *($n \geq 1$) is a sequence of Web Services set and* $S_i \subseteq WS$ *($1 \leq i \leq n$). The predication LayeredlySatisfy:* $(P(WS))^{Nat} \times RQ \rightarrow Bool$ *has the following definition.*

LayeredlySatisfy$((S_1,S_2,\ldots,S_n),r)$ = *true if the following tree conditions holds:*

(a) $getIn(ws)(ws \in S_1) \subseteq getIn(r)$

(b) $getIn(r) \cup (\bigcup_{ws\in S_1} getOut(ws)) \cup \ldots \cup (\bigcup_{ws\in S_{i-1}} getOut(ws))$
$\supseteq (getIn(ws)(ws \in S_i))\ (1 \leq i \leq n)$

(c) $(\bigcup_{ws\in S_{i-1}} getOut(ws)) \cup \ldots \cup (\bigcup_{ws\in S_n} getOut(ws)) \supseteq getOut(r)$

Here, $S_1,\ldots,S_n$ is called an invocation layer sequence (ILS for short) for r and i ($1\leq i\leq n$) is called invocation layer number (ILN for short). Especially, n is called greatest ILN (GILN for short). Obviously, we can get getOut(r) by n layer invocations. According to the definitions above, *FullySatisfy* is the special case of *LayeredlySatisfy*.

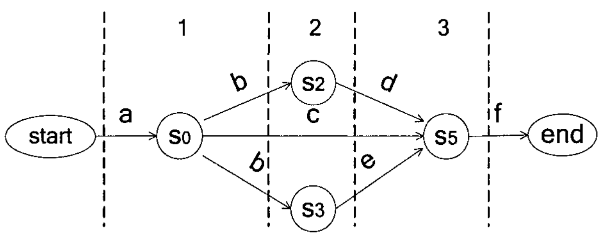


Fig.1. Invocation layer example

For example, in Fig.1, there are four web services s_0, s_2, s_3 and s_5 with $s_0 = (\{a\},\{b,c\})$, $s_2 = (\{b\},\{d\})$, $s_3 = (\{b\},\{e\})$, $s_5 = (\{d,e,c\},\{f\})$, and a service request r with $r=(\{a\},\{f\})$. Obviously, *LayeredlySatisfy* $((\{s_0\},\{s_2, s_3\},\{s_5\}),r)$ stands and the *GILN* is 3.

3. FIXPOINTS THEOREM BASED ALGORITHM

3.1 Algorithm Description

From the analysis above, we can conclude that the key problem of web services composition is how to implement an algorithm to find the web services satisfying the predication *LayeredlySatisfy*. The pseudo code of our algorithm is shown as follows.

Algorithm GetInvocationLayer *(Input: web services corpora WS, service request r; Output: invocation layer layer)*

1) visitedWs $\leftarrow \varnothing$
2) gottenPara$\leftarrow$*getIn(r)*
3) $n \leftarrow 0$
4) $layer[n] \leftarrow \{start\}$
5) While $\neg(gottenPara \supseteq getOut(r))$ *do*
  5.1)$S \leftarrow \{ws \,/\, ws \in WS, ws \notin visitedWs, getIn(ws) \subseteq gottenPara\}$
  5.2) if $S = \varnothing$
    5.2.1) then print "Failure!" and return
  5.3) $n \leftarrow n+1$
  5.4) $layer[n] \leftarrow S$
  5.5) $visitedWs \leftarrow visitedWs \cup S$
  5.6)$gottenPara \leftarrow gottenPara \cup (\bigcup_{ws \in S} getOut(ws))$
6) $n \leftarrow n+1$
7) $layer[n] \leftarrow \{end\}$
8) return

Variable *visitedWs* is a set and used to save the web services that have been visited so far, and variable *gottenPara* is also a set and used to save the parameters that have been available or generated so far. Array variable *layer* is used to save the web services of each invocation layer. Constant *WS* represents a set of all available web services which can be found from a local file system, resources referenced by URIs or provided by a repository such as UDDI. Variable *r* denotes a given web services composition request. Start and end nodes are virtual services that respectively provide require the data from the problem.

At every iteration, some new web services that can be invoked using *gottenPara* are found. At some point, if *gottenPara*$\supseteq$ *getOut(r)*, then it means that using the parameters gathered so far, one can get the desired output parameters in *getOut(r)*, thus finding the web services invocation layers with the least *GILN* satisfying the predication *LayeredlySatisfy*.

3.2 Algorithm Analysis

Theorem 3.2.1 (Termination). GetInvocationLayer will terminate at some point.

Proof. For any given service request $r \in RQ$:

1. If r can be satisfied by some composition of several available atomic web services. Since there are only finite number of web services, and each of iteration of while loop adds only "new" set of web services, the condition of gottenPara $\supseteq$getOut(r) must be satisfied at some point. Then the iteration must end, so the algorithm will terminate.
2. r can not be satisfied by some composition of several available atomic web services. From condition b) of Definition2.5 for LayeredlySatisfy,

we can find that the transition between Layer i-1 and i ($S_{i-1} \Rightarrow S_i$) is a partial order relationship, and the greatest lower bound (glb) is getIn(r) and the least upper bound (lub) is $getIn(r) \cup getOut(r)$. Meanwhile, the transition relationship between invocation layers is monotonic, and therefore, as Knaster-Tarski Theorem [7] implies, there always exists a fix point, ensuring that after this point, gottenPara will not change. That also means that S will not change, then if sentence of Line 5.2) of the algorithm will stand, causing the algorithm to return.

Thereby, inputting any service request, GetInvocationLayer will terminate at some point.

Theorem 3.2.2 (Least GILN) If the input service request can be satisfied by composing existing web services, then GetInvocationLayer can get the ILS $S_1,\ldots,S_n$ satisfying $LayeredlySatisfy((S_1,S_2,\ldots,S_n),r)$ and with the least GILN.

Proof. The former half part of Theorem can be proved by the exit condition of while sentence in Line5. Next, we will proof the latter half part of theorem using counter-evidence. Let $S_1',\ldots,S_m'$ be another ILS of r. That is to say, LayeredlySatify((S_1', S_2',…,S_m'),r) stands and m<n. According to the iteration process of GetInvocationLayer, it will return after the n-th iteration, which is contradicted with the fact that algorithm will return at the m-th iteration. So $S_1,\ldots,S_n$ is with the least GILN.

3.3 Example

For instance, now there is a request r as $r = (\{a\},\{f\})$, and in set WS, a fragment of relevant web services as following: $s_0 = (\{a\},\{b,c\})$, $s_1 = (\{a\},\{g\})$, $s_2 = (\{b\},\{d\})$, $s_3 = (\{b\},\{e\})$, $s_4 = (\{g\},\{h\})$, $s_5 = (\{d,e,c\},\{f\})$, $s_6 = (\{a,h\},\{k\})$

Then the algorithm *GetInvocationLayer* gets the invocation layers as Fig.2.

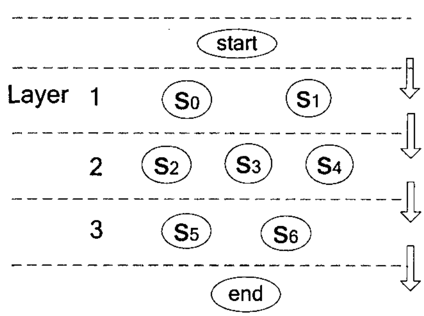


Fig.2. Invocation layers generated by algorithm

4. A* BASED SEARCH ALGORITHM

GetInvocationLayer does not solve the problem of semantic web services composition fully, for the ILS $S_1,\ldots,S_n$ return by it may not be optimal and may include some web services which have no contribution to the service request. However, *GetInvocationLayer* provides a search space where we can find a minimal set of web services contributing to the request. We can get an invocation path by selecting the minimal set of web services from each invocation layer generated by the algorithm. If $LayeredlySatisfy((S_1,S_2,\ldots,S_n),r)$, and at i-th layer, there are m_i web services that can be invoked, then there are $2^{m_i}-1$ search choices at this layer, so there are totally $(2^{m_1}-1)\cdots\cdot(2^{m^n}-1)$ search paths in the search space generated by the algorithm. For instance, in Figure 3, starting from start node, there are 2^2-1=3 ways to invoke subsequent web services :$\{s_0\}$,$\{s_1\}$ and $\{s_0,s_1\}$.Then next, there are 2^3-1=7 ways to invoke: $\{s_2\}$, $\{s_3\}$, $\{s_4\}$, $\{s_2,s_3\}$, $\{s_2,s_4\}$, $\{s_3,s_4\}$and $\{s_2,s_3,s_4\}$. At Layer 3, there are also 3 choices: $\{s_5\}$,$\{s_6\}$ and $\{s_5,s_6\}$. So there are totally $3\times7\times3=63$ paths as Fig.3.The path colored red is our desired one. From this example, we can find the problem that the search space will expanded exponentially, so an effective search algorithm is imperative. In this paper, we propose to use A* procedure [11].

A* procedure is heuristics-based branch-bound search algorithm, with an estimate of remaining distance, combined with the dynamic-programming principle. The heuristics function of A* algorithm is based on the guesses about distances remaining as well as facts about distances already accumulated. It is comprised into two parts as: *u(total path length) = d(already traveled)+ u(distance remaining),*where *d(already traveled)* is the known distance already traveled and *u(distance remaining)* is an estimate of the distance remaining. Since the performance of A* algorithm heavily depends on the quality of the heuristics function, it is important to use the right heuristics to strike a good balance between accuracy and speed.

Definition 4.1 (Heuristics Function) *Given some candidate sets of web services S* ($S\subseteq$ layer[i]) *to visit next at Layer i, we design the heuristics function h as h(S)=d(S)+u(S), where d(S) represents the set of available parameters and u(S) represents the set of remaining parameters of OUT(r). Let output(S) = {s | s is output parameter generated by the visited web services until S in the current search path}. We define d(S) and u(S) as follows:*

$d(S) = |\, getIn(r) \cup output\ (S)\, |$

$u(S) = |\, OUT(r)/output\ (S)\, |$

The pseudo code of our search algorithm base on A*search idea is shown as follows. G is the adjacency-list representation of the graph generated by algorithm *GetInvocationLayer*, whose vertices of layer i are the subsets of variable layer[i] except for ∅ and edges are from one vertex of layer i to each of the next layer i+1 and the root node of G is start.

Algorithm HeuristicsBasedSearch *(Input: service request r, invocation layers layer, heuristics functions d and u; Output: the optimal path π)*

1) Initialize OPEN list

2) Initialize CLOSED list

3) Add start node to the OPEN list

4) while the OPEN list is not empty do

4.1) Get node S off the OPEN list with the lowest h(S)

4.2) Add S to the CLOSED list

4.3) if $d(S) \supseteq getOut(r)$

4.3.1) then return the path from the start node to S according to the function π

4.4) for each $S' \in Adj[S]$ do

4.4.1) $\pi[S'] \leftarrow S$

4.4.2) $d(S') \leftarrow d(S) \cup (\bigcup_{ws \in S'} getOut(ws))$

4.4.3) $h(S') \leftarrow d(S') + u(S')$

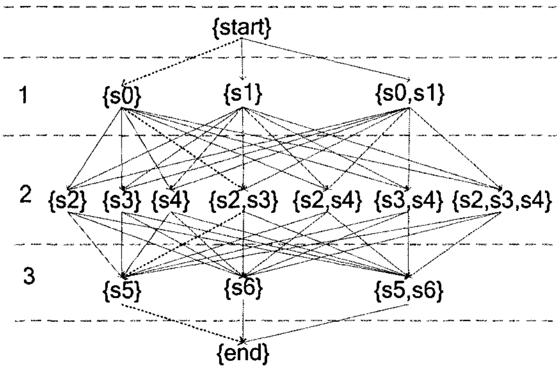


Fig.3. Expanded search space

4.4.4) if S' is on the OPEN list and the existing one is as good or better

4.4.4.1) then discard S' and continue

4.4.5) if S' is on the CLOSED list and the existing one is as good or better

4.4.5.1) then discard S' and continue

4.4.6) Remove occurrences of S' from OPEN and CLOSED list

4.4.7) Add S' to the OPEN list

5) return failure

5. IMPLEMENTATION ISSUES

When implementing the two algorithms above, there are many operations of sets occurring frequently, among which are subset judgment, union, intersection and difference operation. Their implementation efficiency is vital to that of whole algorithm. The key of all these operation is to solve the implementation of membership checking. In this paper, we propose to use Bloom Filter to finish the membership checking operations.

A Bloom Filter is a simple space-efficient randomized data structure for representing a set in order to support membership queries. The space efficiency is achieved at the cost of a small probability of false positives, but often this is a convenient trade-off. Therefore, Bloom Filters have received little attention in the theoretical community. In contrast, for practical applications the price of a constant false positive probability may well be worthwhile to reduce the necessary space. It was invented by Burton Bloom in 1970 [6]. Broder in [3] presents a plethora of recent uses of Bloom Filters in a variety of network contexts, with the aim of making these ideas available to a wider community and the hope of inspiring new applications.

A Bloom Filter for representing a set $S=\{x_1, x_2, ..., x_n\}$ of n elements is described by an array of m bits, initially all set to 0. A Bloom Filter uses k independent hash functions $h_1, \ldots, h_k$ with range $\{1, \ldots, m\}$. We make the natural assumption that these hash functions map each item in the universe to a random number uniform over the range $\{1, \ldots, m\}$ for mathematical convenience. (In practice, reasonable hash functions appear to behave

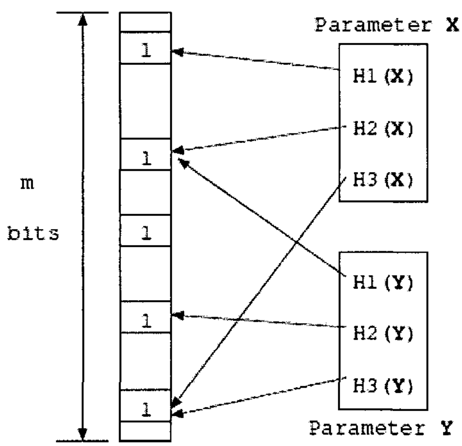


Fig.4. Bloom Filters with three hash functions

adequately, e.g. [4].) For each element $x \in S$, the bits $h_i(x)$ are set to 1 for $i(1 \le i \le k)$. A location can be set to 1 multiple times, but only the first change has an effect. Fig.4 gives Bloom Filters example with three hash functions.

To check if an item y is in S, we check whether all hi(y) are set to 1. If not, then clearly y is not a member of S. If all hi(y) are set to 1, we assume that y is in S, although we are wrong with some probability. Hence a Bloom Filter may yield a false positive, where it suggests that an element y is in S even though it is not. For many applications, false positives may be acceptable as long as their probability is sufficiently small.

The salient feature of Bloom filters is that the probability of a false positive for an element not in the set, or the false positive rate, can be calculated in a straightforward fashion, given our assumption that hash functions are perfectly random. After all the elements of S are hashed into the Bloom Filter, the probability that a specific bit is still 0 is $(1-(1/m))^{kn}$, hence the probability of a false positive in this situation is $(1-(1-(1/m))^{kn})^{k} \approx (1-e^{kn/m})^{k}$, the right hand side is minimized for $k = \ln 2 \times m/n$, in which case it becomes $(1/2)^{k} = (0.6125)^{m/n}$ .In fact, k must be an integer and in practice we might chose a value less than optimal to reduce computational overhead.

6. CONCLUSIONS

This paper studies how web services are composed to provide more complicated services. We propose an algorithm based on the concept of invocation layer and fixpoints theorem, which can be used to get the least invocation layers of candidate web services to satisfy the given service request. Next, we design another search algorithm based on A* procedure to find the best composition ways according to the invocation layers. These two algorithms have been applied to IntelliFlow system prototype developed at CIT to find web services composition setup.

The idea presented in this paper can be extended in future from different points of view. We are interested in solving the problem when specific costs such as time and money are important. Weighted graphs might be a good option to address the problem for these particular issues. As another extension, empowering the approach to support pre-conditions and post-conditions as part of the request is one of our future plans. This will help in specifying more accurate queries and providing more accurate results. The main idea can also be extended to the composition of general software services or even components. If we can somehow extract the required information (inputs, outputs, input-output dependencies) for each available component, the same approach could be used for other types of software services and components as well. This would be consider- ed as another strength of the proposed method.

ACKNOWLEGEMENTS

The work reported in this paper has been funded by the National Grand Fundamental Research 863 Program of China under Grant No.2003AA001023.

REFERENCES

1. B. Benatallah, M. Dumas, Q.Z. Sheng, A.H.H. Ngu, "Declarative Composition and Peer-to-Peer Provisioning of Dynamic web services", Proceedings of the 18th International Conference on Data Engineering (ICDE), 2002.
2. D. Berardi, D. Calvanese, G. De Giacomo, M.Lenzerini and M. Mecella, "Automatic Composition of e-Services", presented at Proceedings of the First International Conference on Service-Oriented Computing (ICSOC), pp. 43-58, 2003
3. A. Broder and M. Mitzenmacher, Network Applications of Bloom Filters: A Survey. Proc. of the 40th Annual Allerton Conference on Communication, Control, and Computing, pages 636-646, 2002.
4. M.V.Ramakrishna, Practical performance of Bloom Filters and parallel free-text searching, Communications of the ACM, 32(10):1237-1239, October 1989.
5. E. Christensen, F. Curbera, G. Meredith, and S.Weerawarana, 2001, "web services Description Language (WSDL) 1.1, W3C Note": www.w3.org/TR/wsdl
6. B. Bloom. Space/time tradeoffs in hash coding with allowable errors. CACM, 13(7):422-426, 1970.
7. A. Tarski. "A Lattice-Theoretical Fixpoint Theorem and its Applications". Pacific J. Math., 5:285–309, 1955.
8. R. Hamadi and B. Benatallah, "A Petri Net-based Model for web services Composition", presented at Proceedings of the Fourteenth Australasian database conference on Database technologies 2003, pp. 191-200, 2003.
9. S. Narayanan and S.A. McIlraith, "Simulation, Verification and Automated Composition of web services", presented at proceedings of the eleventh international conference on World Wide Web, pp. 77-88, 2002.
10. J. Rao, P. Kuungas and M. Matskin, "Application of Linear Logic to web services Composition", presented at Proceedings of the First International Conference on web services (ICWS'2003), pp. 3-9, 2003
11. S. J. Russell and P. Norvig. "Artificial Intelligence: A Modern Approach (2nd Ed.)". Prentice-Hall, 2002.

KNOWLEDGE DICHOTOMY AND SEMANTIC KNOWLEDGE MANAGEMENT

Jiehan Zhou
Technical Research Centre of Finland, Kaitoväylä 1, 90571 Oulu, Finland

Abstract: Today's business is becoming global and knowledge-intensive. This requires business systems capable of knowledge identification, knowledge acquirement, knowledge distribution and knowledge maintenance in terms of universal knowledge understanding. Semantic knowledge management is expected to meet this requirement. This paper presents the fundamentals of semantic knowledge management, including basic concepts, knowledge dichotomy, knowledge solidification modes, and a common semantic knowledge management system.

Key words: knowledge management; semantic knowledge management

1 INTRODUCTION

Modern businesses are knowledge-intensive. The knowledge-intensive business embodies intensive multi-disciplinary knowledge (e.g. a flying car, a photographing mobile phone), intensive product/service decision-making knowledge (e.g. information on supply, geographical allocation, political and culture factors of production resources), intensive product/service implementation knowledge (e.g. distributed collaborative product design, manufacturing and assembly), and intensive product/service knowledge (e.g. global distribution network, multi-cultural customer psychology and aesthetics). This results in growing research on semantic knowledge management with the use of advanced Internet techniques.

Research on knowledge management systems, e.g. CommonKADS (Schreiber 1994), MIKE (Angele 1998), PROTÉGÉ-II (Gennari 2003) is converting the art and craft of knowledge engineering into a real scientific discipline. The current studies such as semantic Web (Semantic Web 2005), ontology engineering (Gómez-Pérez 2004) and semantic search engine (Corby 2002) are being expected to guide knowledge management towards semantic knowledge management.

This paper purposes to present the fundamentals of semantic knowledge management. It is organized in the following way. Section 2 defines basic concepts. Section 3 studies computer-aided knowledge management, including a general knowledge management model, knowledge dichotomy and essentials of computer-aided knowledge management. Section 4 classifies knowledge solidification modes. Section 5 identifies the requirements of semantic knowledge management. Section 6 introduces a common semantic knowledge management system. Section 7 draws a conclusion to the paper.

2 NOTIONS

Data, information and knowledge. Data is an uninterrupted signal. A name, phone number, or contact address of a person is one example. Information is data equipped with meaning. For a car provider, a 'Ford' name is not just a brand of some car object; rather, it is interpreted as an indication of a car-making organization. Knowledge is the whole body of data and information leading a community's activities of making things, for example, the 'Ford' culture or phenomenon. Things include material and immaterial. Knowledge is embedded in things, categorized into explicit knowledge and implicit knowledge (or tacit knowledge). Explicit knowledge can be used for making statements of things with a kind of primitive knowledge, namely knowledge standards. Explicit knowledge usually appears in the form of books, manuals, specifications, standards and methods. Implicit knowledge cannot be described in words, is hard to distribute and exists in an individual's brain and a group's values like brief, experience, know-how, credit and culture.

Knowledge in context. In fact, knowledge depends much on context. The context includes concepts, attribute/values, environment setting, inference rules, and the facts in terms of an action. An action out of context might do things right, but might not do right things. For example, young kids might move chess pieces quickly but wrongly; an expert might do product design decision-making better than a layman; a puzzle fan easily thinks out the answer to a riddle in a reasonable context. Knowledge types include concepts, relations, rules and their instances, which are context-dependent.

An ontology is a shared knowledge standard or knowledge model defining primitive concepts, relations, rules and their instances which comprise a topic knowledge. It can be used for capturing, structuring and enlarging explicit and tacit topic knowledge across people, organizations and computer and software systems. We refer to ontology as knowledge ontology.

Knowledge management. Many studies regard knowledge management as a series of interrelated activities of knowledge identification, acquisition, storage, distribution, reuse, maintenance and development. This paper views knowledge management as two main tasks of knowledge standardization and knowledge instantiation.

Knowledge standardization is to capture knowledge types or knowledge ontology for knowledge instantiation. It can be replaced by the term of ontology engineering. Knowledge instantiation is to exemplify knowledge types or knowledge ontology.

Knowledge management objectives. Knowledge management views knowledge as a structurable resource. Just as with any other resource management, knowledge management aims to provide the resource in a way of 'at the right time, at the right place, in the right form, to the right knowledge worker, with the needed quality and against the lowest possible costs'.

Semantic knowledge management is a method for obtaining knowledge management objectives with a base of knowledge digitalization and knowledge ontology, which is remarkably distinguished from the way of knowledge management in a human brain.

3 COMPUTER-BASED KNOWLEDGE MANAGEMENT

3.1 General knowledge management

There are many knowledge management models. Their common intention is to cover the complete life cycle of knowledge within the organization shown in Figure 1. Typically, the following activities with respect to knowledge and its management are distinguished by many authors(Schreiber 1999).

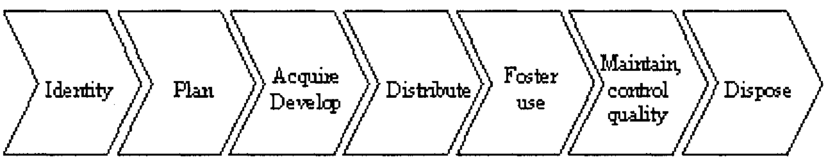


Figure 1. General knowledge management model

- Identify internally and externally existing knowledge
- Plan what knowledge will be needed in the future
- Acquire and /or develop the needed knowledge.
- Distribute the knowledge to where it is needed.
- Foster the application of knowledge in the business processes of the organization.
- Control the quality of knowledge and maintain it.
- Dispose of knowledge when it is no longer needed.

3.2 Nonaka's Knowledge transformation model

Nonaka et al. introduced a knowledge transformation model. Four modes are identified as follows (Nonaka 1995):

- From tacit to tacit knowledge (=socialization): people can teach each other by showing rather than speaking about the subject matter;
- From tacit to explicit knowledge (=externalization): knowledge practices are clarified by putting them down on paper, formulate them in formal procedures, and the like;
- From explicit to explicit knowledge (=combination): creating knowledge through the integration of different pieces of explicit knowledge;
- From explicit to tacit knowledge (=internalization): performing a task frequently leads to a personal state where we can carry out a task successfully without thinking about it.

3.3 Knowledge dichotomy

Knowledge management is a kind of human behavior. Human knowledge management is originated for problem solving and passes through the cycle of human survival and evolution. Human problem solving is pervasive and ubiquitous, from the knowledge discovery to the learning of already existing knowledge. Human knowledge activity might be passive or initiative. Initiative knowledge behavior has an obvious goal, for instance, such as question answering and profit earning; passive knowledge behavior does not have an obvious goal, as in the case of knowledge instillation into a baby. In any case, knowledge behavior takes knowledge standards as a basis. This occurs through a knowledge dichotomy: knowledge consists of instantiation knowledge and standard knowledge. In this way, knowledge management consists of knowledge instantiation, knowledge standardization and knowledge evolution.

- Knowledge instantiation. Human beings usually perform knowledge instantiation by taking knowledge standards as foundation and accepting or rejecting data and facts.
- Knowledge standardization. When a human being is unable to describe the facts or data at hand with existing standard knowledge, knowledge standardization is employed. It begins with a comparison with the existing standard knowledge.
- Knowledge evolution. Knowledge evolution is involved in knowledge instantiation, knowledge standardization and a new formation of knowledge instantiation and knowledge standardization with an enlarged knowledge standard.

According to the above standard-based knowledge management, we transcribe Nonaka's knowledge transformation as follows:

- Knowledge socialization and knowledge internalization. Nonaka's knowledge socialization and knowledge internalization are fermentation processes of knowledge standardization, which enables human beings to describe knowledge in an intuitive manner, such as expression, gesture and emotion.

- Knowledge externalization and knowledge combination. Nonaka's knowledge externalization and combination is a process of knowledge instantiation, which describes and represents knowledge in the guidance of standard knowledge, for instance, signs, languages and symbols.

Take manufacturing knowledge as an example, manufacturing knowledge management consists of manufacturing knowledge standardization and manufacturing knowledge instantiation. Today's manufacturing behavior is mostly carried out with the guidance of standard manufacturing knowledge. Manufacturing knowledge standardization takes in the research and development knowledge on products and production methodologies. For a 'product-out' (just producing the products) enterprise, it is an important factor for it to customize the existing manufacturing knowledge standard to obtain a high business efficiency.

3.4 Computer-aided knowledge management

Knowledge management increasingly plays an important role in human production. The computer is a powerful facility for doing it. A computer-aided human knowledge management model is shown as Figure 2.

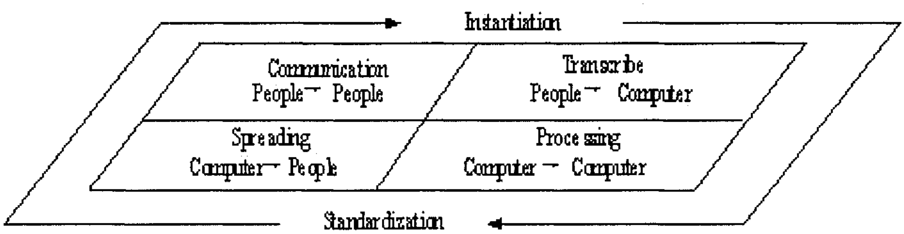


Figure 2. Computer-aided knowledge management

- People to people. It happens when people communicate with each other by natural languages, gestures or expressions. Knowledge standardization takes place gradually in people-to-people knowledge communication. Computer aids for this are not available of this phase
- People to computer. It means people instantiate knowledge with the aid of a computer. An ideal knowledge model facilitates people to exemplify most of the gained knowledge.
- Computer to computer. In which computer stores and processes standard knowledge and instantial knowledge. A new instantial knowledge can be produced by computer inferences.
- Computer to people. Where computer provides people with instantial knowledge and standard knowledge

Consequently, the main functions of computer-aided knowledge management are computer-aided knowledge instantiation and knowledge standardization. Computer-aided knowledge standardization is a process for discovering a new knowledge model, in contrast to an existing knowledge model. This task mainly takes natural language processing and manual knowledge standardization as the basic means. Computer-aided knowledge instantiation facilitates people to fill knowledge models. A computer-aided knowledge query can be viewed as a reverse process of knowledge instantiation.

3.5 Matters in computer-aided knowledge management

Computer-aided knowledge management is an inevitable paradigm co-produced by traditional computer information processing, artificial intelligence and emerging Internet computation. Similar to mechanical devices taking the place of human physical work, computer-aided knowledge management is gradually taking the place of human mental work, for instance, knowledge memory and knowledge discovery. It contains the following four important matters:

- Partly freeing human from mental work. Computer-aided knowledge management keeps people concentrating on creative mental work. In other words, it facilitates people to develop knowledge standards.
- Enlarging the scope of knowledge reusing and sharing. Computer-aided knowledge management not only accelerates binary-coded knowledge spreading, but also distributes semanticized knowledge.
- Reducing knowledge management costs. The cost of standard-based knowledge management will be far lower than non-standard-based type. Computer-aided knowledge management is based on knowledge standards or knowledge models.
- Unifying knowledge processing. People and software process knowledge in conformance to a unified knowledge model, which enhances knowledge evolution.

4 KNOWLEDGE SOLIDIFICATION MODES

Knowledge management accompanies knowledge solidification, namely structuring knowledge. Knowledge solidification has extended over several thousand years since humans came into being. For instance, the simplest knowledge solidification is to remember things with physical brains. Along with the increasing development of computing techniques, it changes profoundly in knowledge solidification. This paper discusses three kinds of important knowledge solidification modes: human brain-based, paper-based and computer-based modes. The computer-based mode can be classified into four types: data structure-based, entity-relation, object-oriented and semantic-based mode. The computer-based mode, paper-based mode and human brain-based mode have their advantages and disadvantages respectively. For example, the accuracy in the computer-based knowledge query is superior to the human brain-based one. However the computer-based is unable to take the place of the human brain in knowledge evolution forever. The semantic-based type in the computer-based mode is more efficient in problem solving than the entity-relation type. They are all developed by these essential ideas of knowledge sharing and semantic-based knowledge modeling. The scope of knowledge sharing mainly distinguishes these methods. Table 1 presents a summary of the three knowledge solidification modes.

Table 1. Summary of knowledge solidification modes

Mode		Knowledge source	Knowledge type	Readability	Under-standability
Brain-based		Knowledge owner	Richest	Only human	Only human
Paper-based		Paper documents	Richer	Only human	Only human
Computer -based	Data structure	Data structure	Application -specific	Programmer -specific	Programmer-specific
	E-R	Database	Enterprise -specific	Intra -enterprise	Intra-enterprise
	O-O	Class base	Programmer -specific	Programmer -specific	Programmer-specific
	Semantics	Knowledge base	Domain -specific	Domain community	Domain community

4.1 Physical brain-based mode

Brain-based knowledge solidification is an effort of obtaining right knowledge in a process of gathering, transcribing and analyzing manufacturing expert's knowledge. According to the expert division in (Schreiber 1999), the three types of academic, practitioner and operator are distinguished. There are knowledge acquisition techniques of interviews, brainstorming and discussions usually used in the brain-based mode. The acquired knowledge is confined to the on-site experts. But there is more to say about the nature of experts that is rooted in the general principles of human information processing. Psychology has demonstrated the limitations, cognitive biases, and prejudices that pervade all brain-based knowledge acquisition. Considering this evidence, it is possible that experts may not have access to the same information when in a knowledge acquisition interview as they do when actually performing the task.

4.2 Paper-based mode

Paper-based knowledge solidification explores the right knowledge by reading, marking up and annotating technical documents. The knowledge source may be in one of the possible forms of messages, text files, paper books, manuals, notes, etc. Unlike individual experts, documents hardly contain very practical know-how acquired through experience. Indeed, they are a consensual view on the domain. Before hard copy-based knowledge acquisition can be taken, the knowledge worker must be sufficiently acquainted with the domain, and the required documents must be accessible.

4.3 Computer-based modes

Computer-based mode refers to that kind of knowledge acquisition in which both an expert's knowledge and document knowledge are formalized in a digital form by means of data modeling. The computer-based mode is categorized as data structure-based, entity-relation-based, object-oriented and semantic-based modes. The borderlines between them are not sharp, because they are relative in terms of context.

- Data structure-based mode. In this mode, knowledge is held in a structured manner that adheres to a well-defined model. This model may be proprietary, however, we are increasingly seeing the appearance of common models that adhere to ISO standards e.g. STEPs(NIST 2005). For example, a geometric wireframe entity is presented by points, lines, and arcs.
- The Entity-Relation-based mode uses enterprise databases as the knowledge source. Entity-Relation (ER) is used as the main method for designing enterprise databases. The ER method views the real world as entities and relations. The basic ER components are entities, relationships, attributes etc.
- Object-Oriented mode. In this, knowledge is written and acquired in terms of real-world objects, classes, subclass/super class, attributes, not internal data structures. This makes knowledge somewhat easier to understand by maintainers and people who have to read the knowledge code.
- The semantic-based mode helps knowledge workers set up complex analyses and structuring of knowledge acquisition. Knowledge users capture knowledge in the context of definitions of concepts, relations and rules, and instances.

5 REQUIREMENTS ON SEMANTIC KNOWLEDGE MANAGEMENT

Many new requirements are proposed for semantic knowledge management as follows:

- Manufacturing knowledge sharing and reusing. Abilities to share and reuse manufacturing knowledge are the primary requirements in developing manufacturing knowledge management system.
- Manufacturing knowledge types. More rich knowledge types are required for manufacturing enterprises to customize production

knowledge due to the transition from 'product-out' and 'market-in' to 'knowledge innovation' in manufacturing business. Current manufacturing knowledge types are flat and specified to applications to a certain extent.

- Knowledge quality. It is important for enterprises to get qualified knowledge. The accuracy plays a key role in weighting knowledge quality. The knowledge accuracy refers to the rightness for an obtained knowledge to the true knowledge (EPISTLE 2005). The two preconditions of improving knowledge accuracy are: knowledge standardization and knowledge instantiation.
- Knowledge cost. Knowledge cost is the total expenditure of knowledge standardization, knowledge instantiation and knowledge evolution (EPISTLE 2005). The cost in human brain-based mode rises quickly during the post-stage of knowledge management, but the cost in semantics-based mode decreases and tends to be steady in the post-stage of knowledge management.
- Knowledge timeliness. The increasingly competitive business requests manufacturing enterprises not only to obtain the accurate knowledge, but also to get it without delay (EPISTLE 2005).
- Knowledge unification. Knowledge unification is the availability of a clear and shared definition for the knowledge. Knowledge unification allows knowledge community to spread and create knowledge quickly.

6 A COMMON SEMANTIC KNOWLEDGE MANAGEMENT SYSTEM

Modern industry communities are respectively driven by knowledge-extensive business. The right knowledge is required for knowledge workers to solve problems rightly, not right now. How to accomplish business knowledge standardization and knowledge instantiation are two key points in realizing semantic knowledge management. Knowledge query could be done by manual means and Internet means. Textbook reading and expert consultation are two examples of the manual one. The manual knowledge query costs much. The knowledge quality is decided by the expert competency and textbook writers. Nowadays, there is an inadequacy in electronic manufacturing documents (e.g. plain and flat Web page) and knowledge search techniques (e.g. plain and flat query interface) for Internet knowledge query. These are incapable for meeting the needs of knowledge timeliness and knowledge accuracy. Knowledge instantiation is another point in modern knowledge management. Human brain-based and paper-based knowledge instantiation are two main ways for knowledge storage, in which it is difficult and costly to convey and renew knowledge.

Conventional knowledge evolution is done by creating new manufacturing concepts, rules, and relations manually.

A common semantic knowledge management approach is a semantics-based knowledge solidification system running over the Internet for managing and developing manufacturing knowledge, which consists of the three functions of knowledge standardization, knowledge instantiation and knowledge query as shown in Figure 3.

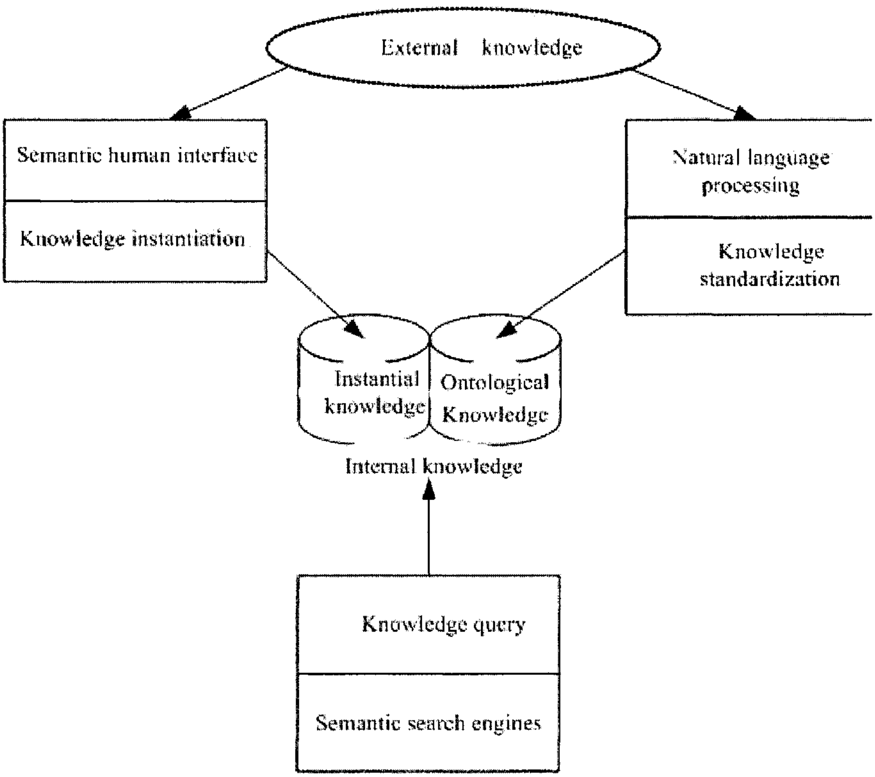


Figure 3. A common semantic knowledge management model

- Knowledge standardization is a computer-aided knowledge modeling process. Take manufacturing knowledge models as an example. There are many standardized manufacturing knowledge models (e.g. STEP (NIST 2005), BPEL4WS (BPEL4WS 2005), EPISTLE (EPISTLE 2005)), which are references for extending manufacturing knowledge model (e.g. new concepts, new rules, and new relations).
- Knowledge instantiation is a computer-aided process of making knowledge instances. Semantic knowledge management provides knowledge workers a semantic human-computer interface for customizing knowledge models. entering a new knowledge instance

entry. If the outward knowledge is capable of being inserted into a knowledge base, it is completely decided by the richness of the existing manufacturing knowledge model.

- Knowledge query is a computer-aided process of acquiring standard knowledge and instantial knowledge. Compared with a plain keywords-based search interface, a semantic knowledge query is advantageous in accurate and proper knowledge query with customized manufacturing knowledge types available.

7 CONCLUSION

Global commercial demands are promoting a growth in the reseach of semantic knowledge management. This paper presented the fundamentals of semantic knowledge management, including basic concepts, computer-aided knowledge management, knowledge dichotomy, knowledge solidification modes, requirements of semantic knowledge management and a common semantic knowledge management system. These fundamentals were successfully applied to implement a semantic manufacturing knowledge management system (Zhou 2004a; Zhou 2004) .

ACKNOWLEDGEMENTS

Thanks to Prof. Rose Dieng for her advices and thanks to ERCIM (the European Research Consortium for Informatics and Mathematics) organization for the financial support.

REFERENCES

Angele, J., et al. (1998). "Developing Knowledge-Based Systems with MIKE." Journal of Automated Software Engineering 5(4): 389-418.

BPEL4WS (2005). Business Process Execution Language for Web Services. http://xml.coverpages.org/ni2003-04-16-a.html

Corby, O., Faron, C. (2002). Corese : A corporate semantic Web engine. 11th International World Wide Web Conference Workshop on Real World RDF and Semantic Web Applications, Hawai, USA.

EPISTLE (2005). Process Industries Data Handover Guide. Part 1 and Part 2. http://www.epistle.ws/

Gennari, J. H. e. a. (2003). "The Evolution of Protege: An Environment for Knowledge-Based Systems Development." International Journal of Human-Computer Studies.

Gómez-Pérez, A., et al. (2004). Ontological engineering. London, Springer Verlag.

NIST (2005). STEP Online Infermation Server. http://www.nist.gov/sc4/

Nonaka, I., H.Takeuchi (1995). The Knowledge Creating Company, Oxford University Press.

Schreiber, G. e. a. (1999). Knowledge Engineering and Management: The CommonKADS Methodology, MIT Press.

Schreiber, T., et al. (1994). "CommonKADS: A Compre-hensive Methodology for KBS Development." IEEE Expert: 28-37.

Semantic Web, W. C. (2005). Semantic Web News and Events. http://www.w3.org/2001/sw/

Zhou J., Dieng R. (2004a). Manufacturing Ontology Analysis and Design: towards Excellent Manufacturing. INDIN'04 2nd IEEE International Conference on Industrial Informatics, Berlin, Germany.

Zhou J., Dieng R. (2004). A Semantic knowledge management system for knowledge-intensive manufacturing. 2004 IADIS International Conference of e-Commerce, Lisbon, Portugal.

SERVICE PORTABILITY FRAMEWORK FOR INTEGRATED COMMUNICATION ENVIRONMENTS

Dmytro Zhovtobryukh[1] and Veikko Hara[2]
[1]Department of Mathematical Information Technology, University of Jyväskylä, 40014 Jyväskylä, Finland; [2]TeliaSonera Corporation

Abstract: Future services must become intelligent to meet the high demands of pervasive computing environments. But until pervasive systems with their ambient intelligence supersede conventional mobile computing environments, it is quite a challenge to incorporate context-awareness and adaptability in services currently available. Such step would bring outstanding flexibility and ubiquity to contemporary mobile computing systems and semantically rich web environments. This paper presents a distinct vision of portable service provisioning which elaborates the concept of a portable service by proposing a dynamic reconfigurable service application design based on context-aware infrastructure support.

Keywords: service portability, service adaptation, network interoperability, context awareness, semantic ontology, industrial Semantic Web environment

1. INTRODUCTION

Current services are designed to operate in specific communication environments. The approach is rather awkward as far as bringing ubiquity into computing and communication to prospective customers is concerned, especially taking into account the growing tendency to integrate modern communication systems. But regardless of whether two adjacent network systems are interconnected or not, the operating service application in the majority of cases should be immediately terminated and reinitiated as the user's terminal crosses the border between the two systems and reassigns its connection to the new one in its range. This is a problem due to the lack of appropriate infrastructure support enabling seamless operation of service application throughout multiple network systems. However, even where it exists, such infrastructure support does not spread on more than two systems. Needless to say that beside apparent performance issues this, beyond doubt, distracts potential users. Therefore it is highly desirable to provide customers with services that roam among interconnected network systems and even various devices, in an effectively transparent and continuous manner. To achieve that, pervasive computing environments, richly endowed with ambient intelligence, would be of great assistance, but unfortunately they are yet a long way from being widespread. Nonetheless, certain steps towards more universal service provisioning and seamless service consumption can be made already now.

What should be dealt with in the first place is the rigidity of services. They should not be oriented towards specific environments, but should be flexible so that they can be consumed by different users, through diverse communication systems and with various devices. Pervasive (ubiquitous) computing[1,2,3] paradigm addresses this issue by decoupling services, applications, devices and users from each other and viewing them as completely independent entities. They are no longer firmly tied together, but have their own functions and objectives and interact with one another when needed. In particular, applications are seen as special entities that perform specialized tasks on users' behalf. Appropriate infrastructure support allows them to be highly customizable and personalized according to users' needs, roam freely between various devices, adapt to changing environmental conditions and be independent of the underlying communication technology. Similar infrastructure support would be a desirable amendment to modern computing systems as well. Not only would it increase service reusability and improve users' perception, but also it would bring current communication standards closer to each other and alleviate the further escalating problem of network interoperability.

This article describes a vision of what is adequate infrastructure support for present-day interoperating communication environments. We propose a reflective context-aware infrastructure for building, rapid prototyping and dynamic adaptation of portable service applications. Some specific details about the proposed service provisioning framework are omitted or not addressed yet and left for further study. However, we believe this article gives a good idea of portable services and of network interoperability problem.

Context-aware computing is nowadays one of the hottest research fields in communications because adaptive intelligent applications are currently in great demand. Numerous researchers and research groups actively develop middleware infrastructures for adaptable context-aware service applications. They seek fresh and robust solutions for next generation computing and networks. However, the majority of proposed infrastructures have limited application due to their orientation on the future communication standards or on highly specific practical implementations, such as smart spaces. For example, Chen and colleagues[4] work on development of smart meeting room system called EasyMeeting that relies on the agent-based context-aware middleware infrastructure Cobra. Gu and colleagues[5] develop an interesting context-aware architecture, which is based on the Open Service Gateway Initiative and is utilizing semantic ontology reasoning, for smart-home environments. Hewlett-Packard's project Cooltown is focused on a Web-based infrastructure for context-awareness[6]. ContextToolkit features a programming approach for modeling and rapid prototyping of context-aware applications[7]. Some other related research activities are described by Chen and Kotz[8]. In contrast to them this article portrays an infrastructure that can be applied to a wide spectrum of contemporary communication environments on a rather wide scale and it specifically focuses on service provisioning and delivery. We feel that the application of our vision to Semantic Web environments is currently one of the most challenging, since semantically rich Web services have to be context-sensitive to become really intelligent.

The article is organized as follows. After this introductory part we present our vision of the Service Portability framework in Section 2. In Section 3 we will speak in detail about service adaptation patterns and introduce the sketch design of our context-aware middleware infrastructure. Section 4 focuses on the perspective of a particular application of the presented vision in an industrial Semantic Web environment. Finally, Section 5 concludes the paper discussing the lessons learned and motivating our future work.

2. SERVICE PORTABILITY FRAMEWORK

The main design goal of the service portability framework is to set up a base for seamless provision of any service through any type of environment. By environment it is meant here a particular combination of physical surroundings and computing environment. The latter is the most important aspect because operational characteristics of each environment mainly depend on the present computing facilities and deployed communication standards. By seamless service provision we mean that services are delivered to users in a continuous and distraction-free manner regardless of any changes that may occur in an environment during an active service session. Though such service provisioning paradigm is quite a challenge, and may be unattainable in practice, the objective is to make services as independent of environments as possible. The main idea is not new: to distinguish between two service instances, one of which is unique and as generic as possible, and the other one a highly specific implementation of the service. No matter how the potential environments differ, it should be ensured that the service is always presented by the unique global instance. This instance should always stay unchanged to ensure that the service remains the same while its specific implementations are customized with respect to the requirements of concrete environments.

In order to achieve this goal, it is necessary to introduce a special service provisioning architecture that would separate the global unique service instance from its actual implementation for a specific environment. (Similar ideas were initially proposed by Banavar and collegues[2]). Such architecture should manifestly adhere to two-phase principle of service provisioning, where the individual phases are virtually independent and concerned with generic and specific service instances respectively.

It is plain from the definition that the two-phase service provisioning comprises two distinct phases. The first phase is *service creation*. It consists of design and deployment of the global instance of the service. Let us call this unique instance a *generic service*. "Generic" here means that on this stage the service definition is devoid of any specific properties related to environments where the service is intended to be used. The closest practical analogue of a generic service is a Web service, which is relieved from low-level details and is advertised by semantic service description that is practically a collection of metadata describing a service. Despite the similarity we intentionally distinguish a Generic service from a Web service to eliminate any association of our conceptual view of a service with its concrete technological realizations for now.

The other phase is *service delivery*. It embodies a construction of a service application from the generic service created during the service creation phase. The service application is already a specific implementation of the service. And since it is aimed for use within a particular environment, all the necessary properties and functionalities are incorporated into it. In other words, every single application of the same service specifically accommodates to the requirements, properties and restrictions of the concrete environment. It must be noted that at this stage, while the application might be altered, the service still remains unaltered. Having been constructed and launched, the application should continuously stay tuned to the requirements until its termination. However, there is a tendency of environmental conditions to progressively change, which makes the process of delivering service applications more complicated.

The concept of service portability implies not only that every generic service may obtain a multitude of differing applications in various environments, but also that during service delivery environments and their properties may dynamically change. As long as such changes render service applications inadequate or even inoperative, the applications should be adapted at run-time not to disrupt pending service sessions.

The function of applications' adaptation is performed by a specific middleware for service portability (as well as by the applications themselves or in collaboration between the applications and the middleware). We propose a context-aware reflective middleware architecture for service portability. Context-awareness plays an important role here as it is a good foundation for application adaptation frameworks and as it allows an even treatment of environmental conditions along with other contexts. Reflectivity is a vitally important property for such middleware, since it allows capturing dynamics, making the architecture viable in a highly dynamic mobile environment.

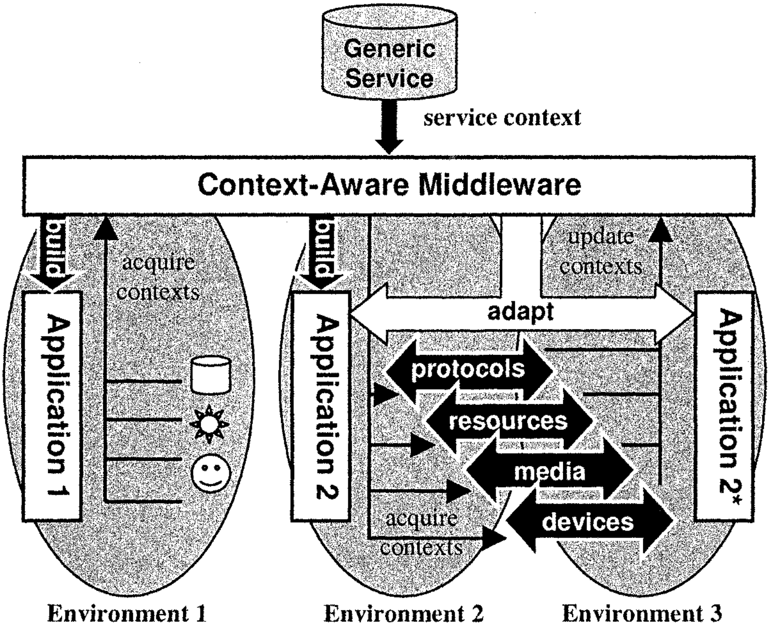


Figure 1. General view of the service portability framework

Figure 1 shows the service portability framework in a general setting. When a user or some application invokes a service from the service provider, the context-aware middleware receives a service context of the corresponding generic service located in the provider's network. The middleware also analyzes various context information acquired from the target environment, and builds a new service application based on the results of the analysis made. As it can be seen from Figure 1, in this way the middleware can produce different service applications (Application 1 and Application 2 in the figure) for different environments from the same generic service. All the necessary information influencing the application being built is obtained from within the context acquired from the target environment. However, the environment, where the application is running, can suddenly change. For example, user may move to another device, or hand over to another environment with different transmission medium and protocol stack, some resources in the environment may vanish or their levels may change. Though this list is far from complete, all these changes (shown in Figure 1 with thick black arrows) essentially mean that the environment has changed. All of them are retrieved by the middleware in the form of context at run-time. At this point the application may turn inoperative, since the environment has changed. To prevent that the middleware has to perform a run-time adaptation of the application (or the application can adapt itself, if it is able to do that). The middleware detects context changes, analyzes them and performs an adaptation procedure on the application.

2.1 Service creation

As it was already noted above, at the service creation stage of service provisioning procedure a generic service is being designed. The obvious objective of this design effort is to supply the service definition with all the necessary data to make the service instance unique, unambiguous and easily interpretable. The generic service instance should be unique in the sense that it should be the only one for the purpose and distinguishable from any other service, i.e. it would not be confused with any other similar service. At the same time the service definition should not be overloaded with any redundant details, which might, in the worst case, turn the service useless for certain environments. A generic service basically contains only metadata about the content of the service and about its operational requirements. Properties related to network standards, protocol stacks, data formats, transmission characteristics, and device capabilities are application-specific and should be implemented within an actual service application. In Web Service Architecture such metadata is called service description[9]. For example, it may provide a link to the service content source, list restrictions on capabilities of end terminal equipment, store some authentication information, etc. Metadata contained in a generic service is to be used during the service delivery stage to construct an appropriate service application which would deliver the service to an end user in the most consistent and reliable way.

Such generic service design is beneficial from several points of view. First of all, it allows maintaining a unique globally available service instance, which is easily recognizable and is not likely to be confused with any other service entities. Secondly, this instance is really generic, which means that it does not possess any implementation-specific properties. Being generic, it does not shrink the range of possible specific implementations of the same service and increases potential reusability of the service. Finally, due to separation of the service and its application, the generic service design allows service porting between diverse environments without any alterations of the service itself.

One more attractive opportunity offered by the generic service design is service composition. In such service provisioning framework two or more services can be jointly delivered by a single application in a rather simple fashion. However, complex interaction of two services at run-time would still present a quite complicated problem.

2.2 Service delivery

Service delivery phase begins with the construction of the service application and ends with its termination. This phase can be logically split into two sub-phases: load-time and run-time. At load time, i.e. build time, the application is constructed from the generic service and is brought into conformity with initial conditions (context) ascertained from the environment. This way, the application is particularly adjusted to operate on top of certain terminal capabilities, protocol stack, resource level, etc. Thus, it is guaranteed that the newly launched application does not fail to perform correctly at startup. At run-time the application runs as normal, but it adapts or gets adapted whenever essential environmental changes occur. Essential environmental changes can be understood as changes in the environment that may influence or even adversely affect the application's performance and operability. Run-time adaptation of applications leads to preservation of service session continuity, which is indispensable for service quality and robustness.

In order to create applications in a rather automatic manner, i.e. without real human designer's intervention, a special framework for application design is needed. An attempt to create a theoretical foundation for such automatic service application design was made in our previous work[10]. In essence, this research work proposes a service reference model for characterization of services with respect to different functional layers, which exist in the system and implement certain functionalities related to services. The main concern of the elaborated model is to create accurate layering of service-related functionalities so that they could be developed relatively independently. What is this needed for then?

Current services are vertical services. This means that they are tightly built into applications that deliver them. They are developed with "all-in-one" principle in mind. All the service-related functionalities from user interface to content rendering are rigidly built into the service instance. The service is seen as a monolith having indivisible structure, which is rather implicit than explicit. Such approach is inflexible in the sense that the service is adaptable only to the extent that is stipulated already at the stage of construction. To illustrate this comment, one can imagine, for example, a GSM short message service (SMS). SMS is designed to exchange text messages, it cannot exchange images and videos. To provide these new possibilities a completely new service called multimedia messaging service (MMS) was created. The restriction to text type of content is implicitly built into SMS, and it cannot be modified to accommodate other types of content. Instead a completely new service needs to be created. Another example is mobile telephony. A user cannot normally cross the border between two different communication networks (by different we mean different network standards) without interruption to the active call, because the application, which delivers calling service to the user, is not designed to be operated on top of a different protocol stack.

An alternative view on the service design focuses on transition from rigid vertical services to horizontal ones. Horizontal services are those provided by the environment. They bring certain system functionality to applications. For example, transport service provides such functionalities as packet transmissions, end-to-end security, traffic management, etc. Applications utilize these functionalities to achieve their specific goals. However, whenever some of the available horizontal services change (like in the last example with GSM telephony), applications are no longer capable of pursuing their own goals, having been intentionally designed to operate on top of a narrow set of horizontal services. In the modern communications world with its strong integration trends such a state of affairs appears increasingly unacceptable. Instead, horizontal services should be effectively used to alleviate the efforts on application design and to make service applications more flexible.

According to our idea, the structure of an application should be modular. It should comprise several functional layers similar to those described by Zhovtobryukh and Kohvakko[10]. Each layer may contain multiple modules that perform specific functions. However, these modules should not be necessarily predetermined at the stage of application construction. Instead, the structure has to be flexible so that modules can be added, removed or substituted at any time. They are to be tied together by some sort of unified interface, which would allow for certain variety of structural alterations. These modules are called *service primitives* and should be mostly provided by the environment where the service application happens to be operating. The role of the modules is not to perform a certain functionality, but to utilize corresponding functionality of horizontal services available in the environment, and to transform this functionality into the form which the application needs.

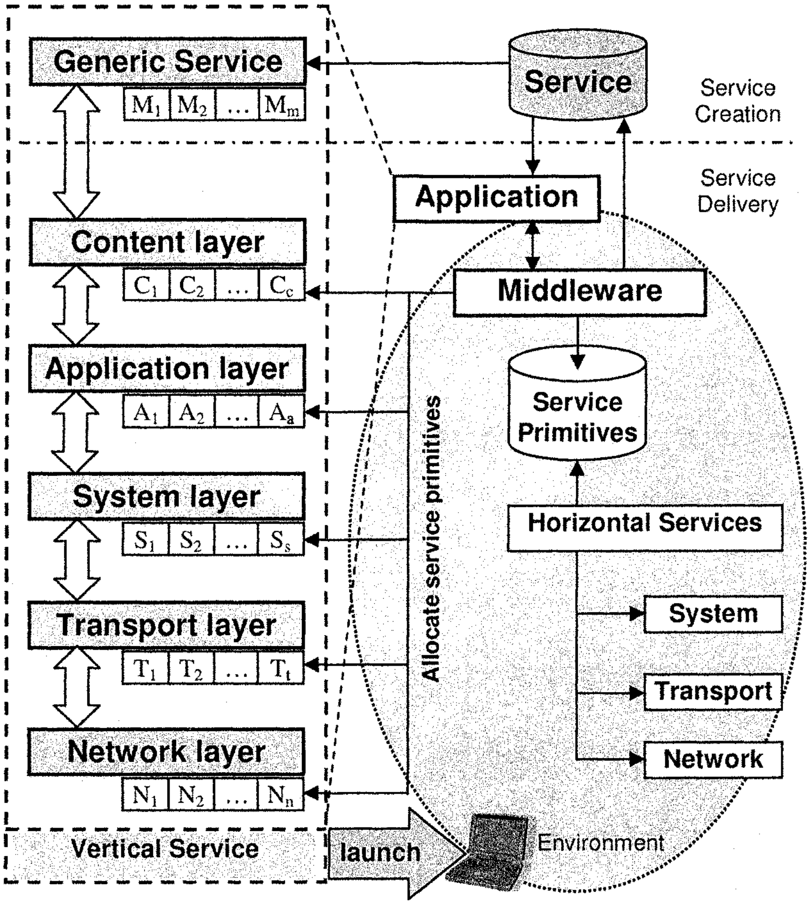


Figure 2. Two-phase service provisioning architecture

Let us explain how this works in a greater detail. Figure 2 depicts the scheme of dynamic assembling of a vertical service application from service primitives available in a concrete environment. To illustrate this scheme in a rather practical way let us use the following scenario.

Steve is at home and is going to leave soon for a lecture which takes place at the university campus. But he knows that a televised football match he has been waiting for the whole day is to start in 5 minutes. Steve takes his handheld device, which has a wireless local area (WLAN) connection to the Internet and through the web interface of the corresponding TV channel's site invokes a video streaming session to watch the broadcast from the match on the way to the university. In the provider's network there is a generic service that provides a connection to the TV broadcast server. The middleware sends a query to this generic service and gets the service context of the video streaming service. This context data contains the address of the TV broadcast server, authentication information for establishing a connection to it and some technical requirements that should be met in order to receive the video streaming service. After that the middleware acquires the context from the environment, namely the device capabilities from the profile of Steve's handheld, characteristics of the active wireless link, traffic management schemes supported by the underlying system architecture, protocol stack used in the WLAN network, etc. Having analyzed this context information, the middleware starts the process of application construction. It locates necessary service primitives with respect to the results of context analysis made and assembles them to get an application. For example, the middleware may obtain video and audio content primitives from the broadcast server to make the application capable of processing and outputting streaming video and audio. It also obtains, from the local environmental repository, the primitive which allows the application to receive packets of streaming traffic type. It is important that the local WLAN network enables horizontal transport service that supports the streaming type of traffic with appropriate Quality-of-Service (QoS) policies. Similarly, the necessary primitives for all the layers shown in Figure 2 are found. Once all the primitives are allocated the vertical service application is finalized and launched on Steve's handheld device. Steve takes his device with him and watches the broadcast on the way to the university.

Thus, a vertical service no longer has a fixed structure. Instead, it is adjusted to concrete environmental requirements by adding appropriate service primitives. This way a multitude of specific implementations of the same generic service can be created.

Portability of a service in an active state, i.e. a service being delivered to a user at that moment, is achieved in a very similar way. At the application's run-time the environment may suddenly change due to the user's switchover to another environment. Having found itself in another type of environment, the application may turn inoperative, which means that some of the currently equipped service primitives are, perhaps, no longer suitable to deliver the service through the new type of environment. Therefore, the application must, accordingly, be adapted to correctly operate in the new environment. The adaptation procedure, which is to be applied in this case, is essentially a substitution of invalid service primitives with valid ones. New primitives are located within the new environment and forwarded to the application. The application reconfigures its structure with the new service primitives and starts operating as normal. The adaptation procedure is primarily controlled by the middleware.

Continuing the previous example, soon after leaving home Steve also leaves the range of his home WLAN connection. His handheld device automatically switches connection to a wide area GPRS network available in the new outdoor environment. The middleware immediately recognizes that changes have taken place in the context and determines that the video streaming application is no longer capable of operating in the new environment. It collects contexts from the new environment, analyzes them and determines which service primitives in the application's structure clash with the new requirements. For instance, the middleware may discover that a different protocol stack is used in the GPRS network. Hence, packets of different format should be received by the client. The middleware contacts a local repository containing service primitives and substitutes all the obsolete primitives within the application with the operational ones. This way the application is being ported to another environment without being terminated and re-launched. In the best of the cases Steve would not even notice the switch between environments. But, generally, some slight distraction period may be evident if the volume of context data is quite significant for fast processing. In the case of non-real-time or asynchronous services strict continuity of a service session is not as important. In these cases packets can be buffered by the middleware somewhere within the system for a later retrieval by the application. The end result of all this is that, without taking any additional actions with his handheld, Steve is watching the TV broadcast while sitting on the bus on his way to the university.

It should be noted that a change of an environment does not necessarily imply that a user has handed over to a different network system. It may also indicate that he has switched over to a different terminal, and an application should immediately "teleport" to the user's new device to preserve the service session from being prematurely disrupted. Some significant changes in the environment, such as, for example, base station crash or resource saturation, will also lead to re-selection of service primitives in order to adapt the application's performance to these new operational conditions.

3. SERVICE ADAPTATION MECHANISMS

The core element of the service portability framework is the context-aware middleware. It is designed to perform actual porting of service applications. The middleware is in charge of service adjustment at application's load-time and of application adaptation at application's run-time. However, application-specific issues cannot be sensibly managed by the middleware. They require application-aware adaptation, in which the middleware assists the application. The types of supported adaptation procedures are described in the subsections below.

In the proposed service adaptation framework all the data which are used as a basis for adaptation are treated as *context* for the purposes of uniformity and simplicity. This would allow collecting and storing all the necessary information in a common repository and inferring adaptation decisions in an easier way.

There are numerous definitions of context in various research fields from artificial intelligence to distributed computing. Some of the most popular definitions can be found in the works of Schilit *et al.*[11], McCarthy[12], Chen and Kotz[8]. However, we believe that the most comprehensive and convenient for our needs definition is provided by Dey[13]: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves". This definition serves well for the proposed architecture because it considers any information related to different entities which are involved in the service provisioning process as context. This view resembles the vision we described in Section 2.

Various contexts are collected by numerous context providers, such as sensors, monitors, and such special sources as user and equipment profiles. Physical contexts, such as location, signal strength, levels of interference, etc., are measured by sensors. Diverse network contexts, e.g. resource levels, are collected by monitors. Service context is retrieved from metadata provided within the generic service. User context can be obtained from user profiles. All the collected contexts are examined by the middleware and stored in the database. The way of modeling context for this infrastructure is a complex issue, especially taking into account such a broad definition of the utilized contextual information. However, we feel that ontological representation of context information is the most flexible and well-developed. Therefore, we assume that context ontologies are used for context modeling. The benefits of this approach will be discussed further in Section 4.

It should be noted that in this article we only consider functional organization of the middleware infrastructure. Architectural layout of its distributed composition is not described yet in detail. However, we will briefly describe the principle of our distributed composition. The proposed middleware architecture consists of three main functional blocks:

- Context Acquisition Proxy
- Context Manager
- Context Reasoning Engine

The middleware manages three data repositories:

- System Contexts
- Context Model
- Service Primitives

Context Acquisition Proxy controls the population of context providers. It collects readings from sensors/monitors at their report rate and sends updates to the context manager. Acting as a proxy, this facility filters received readings. It discards all inessential changes in monitored contexts using certain implicitly given criteria for their estimation.

Context Manager is responsible for maintaining the context model in a consistent state by updating it promptly with essential context changes. It also controls context information exchange between different architectural entities and manages context dissemination procedures.

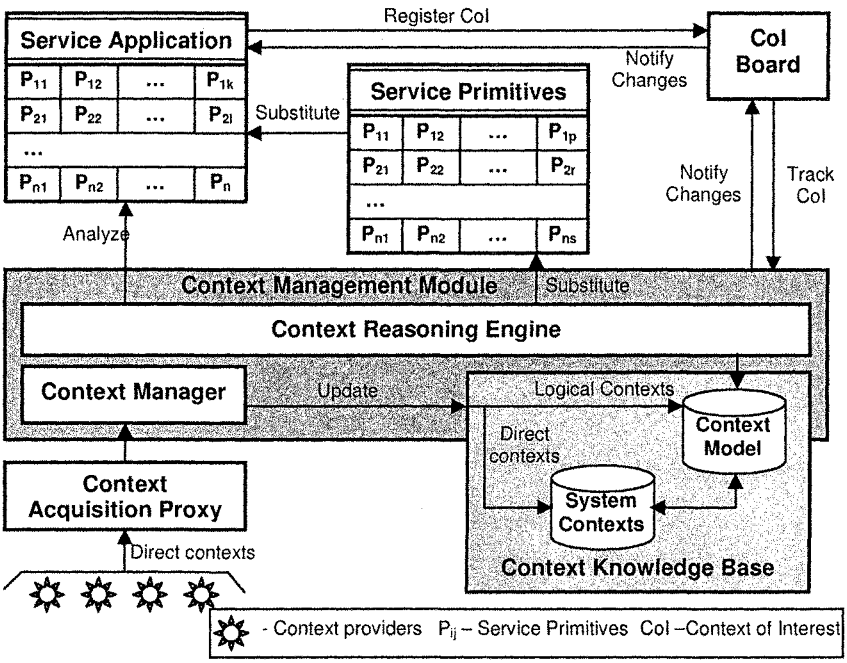


Figure 3. Service adaptation framework

Context Reasoning Engine is the brain of adaptation. Its apparent role is to perform reasoning on the context. The engine checks for changes in the context knowledge base looking for possible conflicts between the application and system contexts. By conflict it is referred to a certain discrepancy between the application's operational properties and the current environment parameters. This discrepancy would form a hindrance for the application to operate properly in the observed conditions and must, therefore, be resolved during the adaptation process. Whenever the context reasoning engine determines such a collision, it starts reasoning on the base of the context model trying to deduce a satisfactory solution for the detected conflict. The reasoning procedure may or may not cause a launch of an adaptation procedure. Adaptation, as an ultimate way of conflict resolution, usually results in a substitution of application's service primitives that clash with occurring contextual conditions. On basis of the obtained reasoning decision the engine selects new primitives from the corresponding repository and instructs the application to substitute them (see Figure 3).

System Contexts Repository is a relational database that stores the whole variety of context variables monitored in the system and their current values.

Context Model Repository contains an ontological model of contexts present in the environment. It may express both physical attributes and logical properties of those contexts. The model is capable of formalizing complex relationships between different contexts. It is maintained in the actual state and gets updated as necessary by the context management facility.

Service Primitives' Repository stores an array of service primitives compatible with a current environment. The primitives are built in correspondence with the horizontal services that the environment provides. These primitives are basically pieces of executable program code. They are shaped as program modules and can be linked to each other through a unified interface. The primitives are constructed by special middleware services, which are out of the scope of this paper. If some substantial changes occur to the environment and influence its horizontal services, the corresponding primitives need to be removed from the repository and new primitives added. The service primitives' repository is used in the adaptation procedure, whenever the middleware decides that some primitives of a certain application are not fit to operate in a current environment. In this case, the middleware makes a decision about which primitives should be selected from the repository to substitute the application's obsolete ones.

From the architectural viewpoint, the middleware is composed in a distributed fashion. The main architectural entities are *Context Management Module* that comprise context manager and context reasoning engine facilities, and *Context Knowledge Base* that consists of context model and system contexts repositories. These two modules are centralized for a certain locality in an environment. Context acquisition proxies are allocated locally for groups of context providers present in certain parts of an environment. Data exchange between local proxies, central management modules and context-aware applications is based on a certain protocol. A survey or any justification of such protocols is beyond the scope of this article. The operation of the middleware architecture is synchronized by means of event triggering system, which allows capturing unexpected situations in contexts.

The operation of the described middleware architecture depends on what type of adaptation procedure is being currently accomplished. The following subsections describe some possible adaptation procedures and specifics of the middleware operation for each of them.

3.1 Background adaptation

Background adaptation is an adaptation procedure that is entirely handled by the middleware without the application's assistance. The application under adaptation may not even be aware of the adaptation performed on it.

This is a default type of adaptation. The middleware always tries to perform it first, and only if the detected problem cannot be resolved by the middleware, it would proceed to other adaptation procedures.

Background adaptation is generally utilized when inappropriate service primitives of the application belong to the layers that correspond to the horizontal services provided by a local environment. For example, if a user has made a handover to another network system which uses a different protocol stack for transmitting packets, or different QoS framework are utilized, then certain service primitives on lower layers of the service reference model may appear inappropriate for a new environment and consequently turn the application inoperative. The middleware is capable of detecting and settling such a problem by its own strength, since transport and lower system layers are transparent to the application. So, the middleware reassigns the primitives of transport and network layers imperceptibly to the application.

The role of the context reasoning engine is to analyze the configuration of the application, and the service context, and to detect which service primitives of the application are obsolete. After that the engine assigns, to the application, new service primitives from the local service primitives' repository.

3.2 Application-aware adaptation

Application-aware adaptation is an adaptation procedure in which the application is aware of the adaptation being made and makes the final decision on how it is to be adapted. This type of adaptation is carried out by the application with a possible assistance of the middleware.

Application-aware adaptation usually takes place when inappropriate service primitives of the application reside on the layers that are not related to a local environment but to the service itself. Metacontent, content and application layers[4] obviously belong to this category. For example, after Steve's handover to a GPRS environment on his way to the university campus it may appear that the system cannot accommodate the quality of video streaming service due to overall scarce bandwidth. The middleware has already failed to satisfy the requirements of the service by performing background adaptation. Therefore, the adaptation has to be made by the application to reduce the quality of the delivered service. In the above example, the application should follow the recommendation of the middleware and stop using the video content primitive, which is too resource-demanding, and keep the audio content primitive only in order to deliver the TV broadcast in an accurate and error-free manner.

This type of adaptation cannot be handled by the middleware on its own because the service primitives, which need to be substituted, are not transparent to the application and comprise its core functionality. Furthermore, new primitives for substitution cannot be retrieved from a local environment in this case because of their absence in the local service primitives' repository. They can only be provided to the application in advance at load-time (or retrieved from the service's origin), so that the application could adapt itself in critical situations. The role of the context reasoning engine here is to detect the conflict, and to attempt to find a solution on its own - after the failure of the background adaptation to analyze what kind of application adjustment is required for a successful solution - and finally to propose a deduced solution to the application for a subsequent accomplishment.

Besides that, the application may be given some adaptation strategies at the construction stage. It may be required to adapt its behavior during its operation with respect to certain changing factors. If the application can perform self-adaptation, it is usually called *adaptive*. The proposed middleware architecture facilitates the support for adaptive applications by providing them with a possibility to register their contexts of interest (CoI) within a special tracking facility called CoI board. When an application registers its CoI within the CoI board, necessary contexts start being tracked by the middleware (if it is technically possible) and the changes are reported directly to the CoI board. The board then notifies the application about the changes in its CoI, and the application can adapt its behavior as necessary.

The benefit of such a framework is in that the application may register quite complex contexts, which cannot be measured directly, but only inferred on the base of the context model. Such complex contexts are deduced by the context reasoning engine, thus relieving the application from sophisticated computations and giving it the possibility to adapt itself almost effortlessly.

3.3 User-aware adaptation

User-aware adaptation is the most complicated type of adaptation. It implies that the final adaptation decision is made consciously by the user because neither the middleware nor the application succeeded to provide a satisfactory service quality level by their own strength.

Nevertheless, all the work to find an appropriate adaptation decision is done by the middleware and the application. In case they succeed to find any reasonable solution, they propose it to the user, who finally decides what to do. Such a solution, if it can be found, is called a *corrective action*[3]. By performing this corrective action the user ensures the required level of service quality.

Let us use the previous example to illustrate this idea. After Steve's handover to a GPRS network system it appears that the available bandwidth is too scarce for a satisfactory quality of video streaming service. The middleware fails to solve the problem due to a low physical capacity of wireless links. The application fails to reduce the quality of the delivered service, because the video component of the broadcast appears critical at the moment (Steve manually set in advance the application's options to indicate that video would be critical to display). However, having analyzed the system and user contexts, the middleware and the application find a corrective action soon after Steve gets out of the bus: if he moves to the nearest lobby, which is situated 30 meters away from his current location, Steve will get an acceptable quality level of video transmission, because the lobby is equipped with a WLAN hot-spot.

In principle, the middleware is capable of deriving corrective actions without any assistance of the application. However, only the application is able to communicate the corrective action that is found to the user.

4. APPLICABILITY TO INDUSTRIAL SEMANTIC WEB ENVIRONMENTS

The service portability framework is described in the previous sections in a rather broad way lacking any details about possible applications of the presented vision. The main intent for such broad description is to show that the framework does not in principle depend on specific environments and/or services.

We believe that the application of the described approach to industrial Semantic Web environments is currently the most suitable and justified way to go about. First of all, the Web Service Architecture[9] does itself provide flexible support for two-phase service provisioning. It can be easily seen that our view on service provisioning presented in section 2 resembles Web Service provisioning framework with minor amendments. These differences should not be applied to web services themselves, but can be realized within the middleware infrastructure.

Another technical motivation for this approach is the use of Semantic Web ontologies for modeling context. There are several major approaches to context modeling[14], and they have their own strengths and weaknesses. Among these approaches Ontological context modeling is currently the most reasonable method due to the following properties of ontologies:

- high flexibility and manageability
- possibility for distributed composition
- capturing incomplete and ambiguous information
- high level of formality
- applicability to existing environments

Furthermore, some of the context sources in a real enterprise environment may already provide contextual information in an ontological form (e.g. user and equipment profiles, service descriptions), which makes it even more logical to use ontological approach for modeling the rest of the context.

Finally, Semantic Web ontologies (in particular, those in the OWL format) provide means not only for context modeling but also for contextual reasoning, which significantly alleviates the implementation efforts for building context reasoning engines. Ontological approach to context modeling has already justified itself within several similar research works.[5,15]

From a practical viewpoint, the described infrastructure for service portability is easier and more reasonable to implement and deploy within a single enterprise network rather than on a wide scale for public communication systems.

A typical service provision scenario for an industrial Semantic Web environment is presented in Figure 4. Service Provider and Service Requestor are generic entities that represent the owner of a Web Service and its consumer respectively. These entities can express relationships of business-to-business type as well as business-to-customer type. The process of service discovery is out of the scope at the current moment.

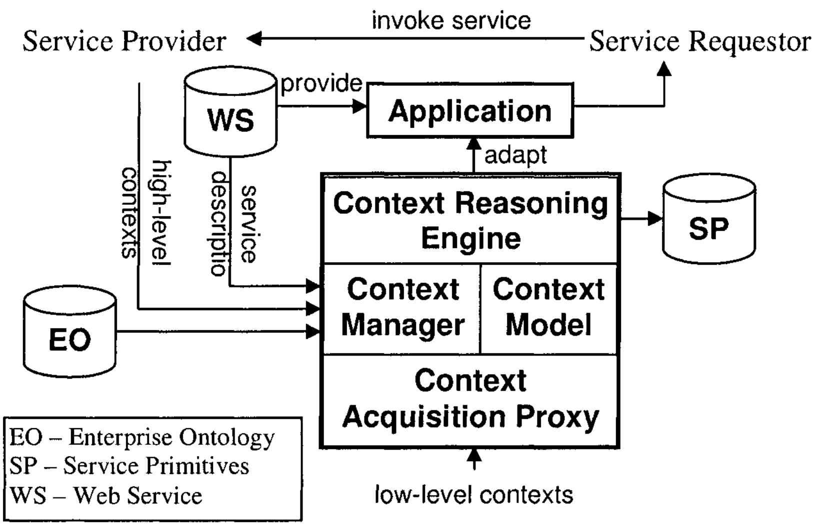


Figure 4. Service provisioning scenario in industrial Semantic Web environment

This scenario differs from the generic case illustrated earlier in the paper is in that high-level contexts are already represented in the ontological form within the enterprise system. The Web Service corresponds to the notion of Generic Service, and the service description, advertised by it, is a concrete instance of service context or meta-context. Low-level contexts, such as time, spatial coordinates, connection characteristics, etc. are received as usual from low-level context providers and transcribed to the ontological view by Context Management Module with respect to the available ontology structures in the system Enterprise Ontology. Enterprise Ontology describes the entire system in a comprehensive manner, annotating properties of its elements and relationships between different entities within the environment.

Let us consider the following example. As soon as Steve arrives to the university campus he realizes that he does not know the place where the lecture is being held. He browses to the university's website and invokes the location-based campus guidance service. He submits a query for a lecture location to the service and gets an answer which indicates that the lecture will be held in the campus building owned by a special unit of an industrial company collaborating with the university. This unit has its own enterprise environment covering the building. As soon as Steve enters the building he is classified by the internal security system as a participant of the visiting lecture and authorized to use certain enterprise services available within the system. At this point the guidance application is adapted to operate on a possibly new technological base present in the enterprise network (new wireless access standards such as WLAN and Bluetooth, new protocol stack, QoS frameworks, etc.) and to use internal location-based Web service providing visitor guidance inside this particular building. To do this the middleware acquires the service description of the corresponding web service, analyzes the context model of the environment, and the local user profile, performs reasoning and finally extracts necessary service primitives from the environment to properly re-configure the application. Following the instructions on the screen of his handheld Steve takes an elevator to the third floor. The guidelines provided to Steve are the result of a context reasoning procedure performed by the new context-aware application and based on Steve's location inside the building. Steve's location is its specific "context of interest", which is tracked by the middleware on the application's demand. The path is constructed and locations are identified with respect to the enterprise ontology, which exists in the system and defines such relationships as, for example, "containment" to enable reasoning about location.

Bringing context on top of semantics in Web Services is an attractive feature of the framework that would allow Web Services to be even more flexible and intelligent. Notice also that the framework has potential to deal with the problem of Web Service composition by using context-awareness for customization of composite services.

5. CONCLUSIONS AND FUTURE WORK

In this article we described our vision on how to bring context-awareness into modern computing and communication environments. The most important thing about the framework presented is that it not only makes context-aware services available to current mobile users, but also proposes a featured context-aware approach to build up interoperability between today's communication systems.

The Service Portability framework is based on the idea of decoupling service applications from actual services. Such approach simplifies service design, since services no longer require to be endowed with application-specific details. This, in its turn, increases service reusability, allowing any service to be consumed through any type of environment without being modified or reissued. The portability of services is achieved by introducing adaptable service applications that can be reconfigured. These applications, instead of services, are adapted whenever environment undergoes any significant change. Such principle allows hiding any unnecessary details from service creators and managing run-time adaptation locally with the particular service application. Sophisticated efforts on remote service adaptation are not needed and service scalability is preserved. To manage the framework we specify a context-aware infrastructure which captures dynamic environmental changes in an efficient manner and provides reconfiguration and adaptation mechanisms for service applications. It utilizes horizontal services provided by environments to build service applications in a more pertinent way and as a result makes service concept more flexible and open.

Although the general principle of how the described infrastructure operates is expounded in the paper and clear motivation for it is given, there is a number of issues to be addressed yet. These include a programming model for reconfigurable applications with a modular structure, distributed composition of the architecture, details of context modeling approach, maintenance of the distributed context model, context dissemination between the parts of the system, etc. Before we can seriously speak about the architecture's feasibility and applicability to modern environments, these questions should be answered, and it is these questions that guide us in our further research work.

REFERENCES

1. M. Weiser, The Computer for the 21st Century. *Scientific American*, 1991
2. G. Banavar, J. Beck, E. Gluzberg, J. Munson, J. Sussman, and D. Zukowski, Challenges: An Application Model for Pervasive Computing, *Proceedings of 6th ACM Mobicom*, 2000.
3. M. Satyanarayanan, Pervasive Computing: Vision and Challenges, *IEEE Personal Communications*, pp.10-17, August 2001.
4. H. Chen, T. Finin, A. Joshi, L. Kagal, F. Perich and D. Chakraborty, Intelligent Agents Meet the Semantic Web in Smart Spaces, *IEEE Internet Computing*, October 2004, pp. 2-12.
5. T. Gu, H. K. Pung, and D. Q. Zhang, Toward an OSGi-Based Infrastructure for Context-Aware Applications, *IEEE Pervasive Computing*, October-December 2004, pp. 66-74.
6 T.Kindberg and J.Barton, A Web-based Nomadic Computing System, *Computer Networks*, Vol. 35, No. 4, pp. 443-456.
7 A.K. Dey, D. Salber, and G.D. Abowd, A Conceptual Framework and Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications, *Human-Computer Interaction*, Vol. 16, No. 2-4, 2001, pp. 97-166.
8. G. Chen and D. Kotz, A Survey of Context-Aware Mobile Computing Research, Technical Report TR2000-381, November 2000.
9. Web Service Architecture, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/, (last retrieved on 13.05.2005)
10. D. Zhovtobryukh and N. Kohvakko, Service Reference Model for Modern Communications, *In Proceedings of the Fourth International Network Conference*, edited by S.M.Furnell, and P.S.Dowland, 2004, pp. 221-228.
11. B. Schilit, N. Adams, and R. Want, Context-Aware Computing Applications, *In Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pp. 85-90, 1994.
12. J. McCarthy, Notes on Formalizing Context, *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp.555-560, 1994.
13. A.K. Dey, Understanding and Using Context, *Personal and Ubiquitous Computing*, Vol. 5, Springer-Verlag, 2001, pp. 4-7.
14. T. Strang and C. Linnhoff-Popien, A Context Modeling Survey, UbiComp 2004 Workshop, 2004
15. H. Chen, T. Finin, and A. Joshi, An Ontology for Context-Aware Pervasive Computing Environments, *ACM Knowledge Engineering Review*, pp. 197-207, Vol. 18, Issue 3, September 2003.
16. D. Zhovtobryukh, Integration Issues in Communication Environments, *In Proceedings of the 1st International Workshop on Ubiquitous Computing IWUC 2004*, 2004, pp. 28-37.
17. A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, The Anatomy of a Context-Aware Application, *Wireless Networks*, March/May 2002, V.8 No. 2/3, pp.187-197.