ESWC06

Budva, Montenegro
11 - 14 June 2006

European Semantic Systems Initiative

3rd European Semantic Web Conference 2006

Proceedings of

Posters and Demos

3rd European Semantic Web Conference

Budva, Montenegro 2006
http://www.eswc2006.org

Posters and Demos

At the 3rd European Semantic Web Conference, Budva, Montenegro 2006
http://www.eswc2006.org

ESWC is the primary European conference on topics related to the Semantic Web. It covers all aspects of Semantic Web research and development. Topics of interest include but are not limited to:

- Ontology Management (e.g. creation, evolution, evaluation)
- Ontology Alignment (e.g. mapping, matching, merging, mediation and reconciliation)
- Ontology Learning and Metadata Generation (including e.g. HLT and ML approaches)
- Multimedia and Semantic Web
- Semantic Annotation of Data
- Semantic Web Trust, Privacy, Security and Intellectual Property Rights
- Semantic Web Rules and Query Languages
- Reasoning on the Web (e.g. scalability, fuzziness, distribution)
- Searching, Querying, Visualizing, Navigating and Browsing the Semantic Web
- Personalization and User Modelling
- User Interfaces and Semantic Web
- Semantic Grid and Middleware
- Semantic Web Services (e.g. description, discovery, invocation, composition)
- Semantic Web-based Knowledge Management (e.g. Semantic Desktop, Knowledge Portals)
- Semantic Web for e-Business, e-Culture, e-Government, e-Health, e-Learning, e-Science
- Database Technologies for the Semantic Web
- Data Semantics and Web Semantics
- Semantic Interoperability
- Semantic Web Mining

Besides the scientific and industrial paper track, ESWC asks for poster and demonstration contributions that will be presented in a special session during the conference.

For the poster and demo session we were looking for contributions whose nature make them less suited for submission to the official paper track. In particular, we ask for contributions of the following kind.

- **Late-breaking and speculative results:** Significant and original ideas with promising approaches to resolve open problems in Semantic Web research that are in an early stage and have not been verified and tested sufficiently to meet the requirements of a scientific publication.
- **Descriptions of system demonstrations:** Descriptions (preferably accompanied by demonstration) of new systems that use Semantic

Web technology to solve important real world problems. We are also looking for software infrastructure supporting the development of systems that use Semantic Web technologies.

- **Projects and initiatives:** Descriptions of the objectives and results of ongoing projects and initiatives. The aim is to provide an overview of ongoing work in the area of the Semantic Web.

We received 69 submissions covering almost all aspects of Semantic Web research. All submissions were reviewed regarding their suitability for the demo and poster session by an internal program committee. We were able to accept 51 of these submissions for presentation at the demo and poster session held on Monday the 12th of June 2006. The high number of 20 accepted system demonstrations shows that Semantic Web technology is leaving the labs and starts delivering technologies and applications that have impact on the future development of the Web. But also the growing number of poster submissions is an indication for challenging ideas from which the Semantic Web can benefit. Providing a forum for showing these developments is an important part of a Semantic Web conference and this demo and poster session tries to strengthen this aspect.

Holger Wache
(Chair)

Acknowledgements

We would like to thank Christen Ensor and Ilona Zaremba for supporting the poster and demo session and the members of the program committee listed below for helping us to select contributions.

- Andreas Abecker (FZI Karlsruhe, DE)
- Jürgen Angele (Ontoprise GmbH, DE)
- Richard Benjamin (ISOCO, Spain)
- Jos de Bruijn (DERI Innsbruck, AT)
- Paul Buitelaar (DFKI Saarbruecken, DE)
- Steve Cayzer (HP, GB)
- Orscar Corcho (University of Manchester, UK)
- Peter Dolog (University of Hannover and L3S, DE)
- Aldo Gangemi (CNR, IT)
- Volker Haarslev (Concordia University, CA)
- Pascal Hitzler (University of Karlsruhe, DE)
- Eero Hyvönen (University of Helsinki, FI)
- Ruben Lara (TIF Tecnología, Spain)
- Riichiro Mizoguchi (Osaka University, JP)
- Enrico Motta (The Open University, UK)
- Lyndon Nixon (Free University Berlin, DE)
- Terry Payne (University of Southampton, UK)
- Aleksander Pivk (J. Stefan Institute, SL)
- Pavel Shvaiko (University of Trento, IT)
- Michael Sintek (DFKI Kaiserslautern, DE)
- Heiner Stuckenschmidt (University of Mannheim, DE)
- Vojtech Svatek (University of Economics, CZ)
- Sergio Tessaris (Free University Bozen, IT)
- Anita de Waard (Elsevier, NL)
- Paul Warren (British Telecom, UK)

Table of Contents

Demos

Posters

Demos

SmartWeb: Mobile Access to the Semantic Web

A. Ankolekar, P. Cimiano, P. Hitzler, M. Krötzsch, G. Ladwig, H. Lewen, D. Oberle, R. Studer
AIFB, Universität Karlsruhe, Germany

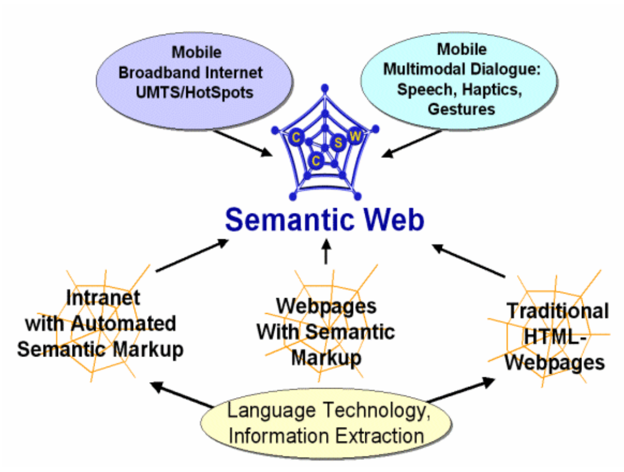
M. Kiesel, M. Sintek
DFKI Kaiserslautern, Germany

ABSTRACT

We present the SmartWeb Demonstrator for multimodal and mobile querying of semantic resources and the open WWW. The end-user interface consists of a Pocket Data Assistant which accepts written or spoken questions as input and delivers answers based on a multitude of resources including a semantic knowledge base, semantically annotated online web services, and semi-automatically created knowledge from text-based web pages. If answers cannot be found using these structured resources, then the system returns answers based on linguistic query-answering techniques on the open WWW.

1. INTRODUCTION

Recent progress in mobile broadband communication and semantic web technology is enabling innovative internet services that provide advanced personalization and localization features. The goal of the SmartWeb (http://www.smartweb-project.de) project is to lay the foundations for multimodal user interfaces to distributed semantic web resources and services on mobile devices. The SmartWeb consortium brings together experts from various research communities: mobile services, intelligent user interfaces, language and speech technology, information extraction, and semantic web technologies.



SmartWeb is based on two parallel efforts that have the potential of forming the basis for an advancement of the web. The first effort is the Semantic Web, which provides the tools for the explicit markup of the content of web pages. The second effort is the development of semantic web services which results in a web where programs act as autonomous agents to become the producers and consumers of information and enable automation of transactions.

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 11th - 14th June, 2006

The appeal of being able to ask a question to a mobile internet terminal and receive an answer immediately has been renewed by the broad availability of information on the web. Ideally, a spoken dialogue system that uses the web as its knowledge base would be able to answer a broad range of questions. SmartWeb exploits the machine-understandable content of semantic web pages for intelligent question-answering as a next step beyond today's search engines. Since semantically annotated web pages are still very rare due to the time-consuming and costly manual markup, SmartWeb is using advanced language technology and information extraction methods for the automatic annotation of traditional web pages encoded in HTML or XML.

SmartWeb provides a context-aware user interface, so that it can support the user in different roles, e.g. as a car driver, a motor biker, a pedestrian or a sports spectator. One of the demonstrators of SmartWeb is a personal guide for the 2006 FIFA world cup in Germany, that provides mobile infotainment services to soccer fans, anywhere and anytime, using a PDA as user-interface. We will present this demonstrator at the conference.

The academic partners of SmartWeb are the research institutes DFKI (consortium leader, Prof. Dr. Wolfgang Wahlster), FhG FIRST, and ICSI together with university groups from Erlangen, Karlsruhe, Munich, Saarbrücken, and Stuttgart. The industrial partners of SmartWeb are BMW, DaimlerChrysler, Deutsche Telekom, and Siemens as large companies, as well as EML, Ontoprise, and Sympalog as small businesses. The German Federal Ministry of Education and Research (BMBF) is funding

the SmartWeb consortium from 2004 to 2007. We acknowledge the funding under project number 01 IMD01 B.

2. MULTIMODAL RECOGNITION AND MODELING

Access to SmartWeb is either gained on the field via a PDA/Smartphone (UMTS) with a server-based speech-recognizer or in a mobile car-scenario via a built-in speech-recognizer in the car. On the PDA, the interface is supplemented with multimodal input, e.g. with a pen. In addition, a camera monitors the face and recognizes whether the user is addressing the system.

For the development of a mobile multimodal dialogue assistant in SmartWeb that is usable in open domains and thematically wide-ranging areas, spoken language is the central mode of communication. In communications that are situation dependent and technically allow the use of the whole range of multimodal functionalities, the phonological output of the dialogue assistant using different output modalities – for example music as another form of acoustic output or visual (text, graphic, picture, video) and haptic representations – has to be spatially and timely synchronized.

3. ONTOLOGICAL INFRASTRUCTURE

The SmartWeb project comprises the definition, implementation, and application of ontologies for various parts of the system. Using ontologies enables the formalization of concepts that are understood and accepted by a wide user basis. They lay the foundation for the dialogue with the user as well as for the flexible communication between applications from various, wide-ranging areas. The ontologies also enable important basic tasks such as the formulation of structural queries and inferencing.

The SmartWeb integrated ontology consists of several domain ontologies which are aligned by means of an adaptation of SUMO and DOLCE. The domain ontologies describe sport events, navigation information, multimodal interaction discourses, multimedia data, and linguistic information. The purpose of the domain ontologies is not only to provide accurate answers to queries. They are also used in other parts of the system, e.g. for semantic annotation of web services, for modelling the multimodal interaction between the user and the SmartWeb system, and for handling linguistic information about objects and classes.

Different ontology representation languages are used within different subsystems. RDFS is being used dominantly, in particular for the domain ontologies. OWL is being used for representing more complex information, e.g. for the foundational ontology which was utilized for aligning the domain ontologies. Reasoning support and intelligent knowledge processing is provided by the Ontobroker system which is based on F-Logic.

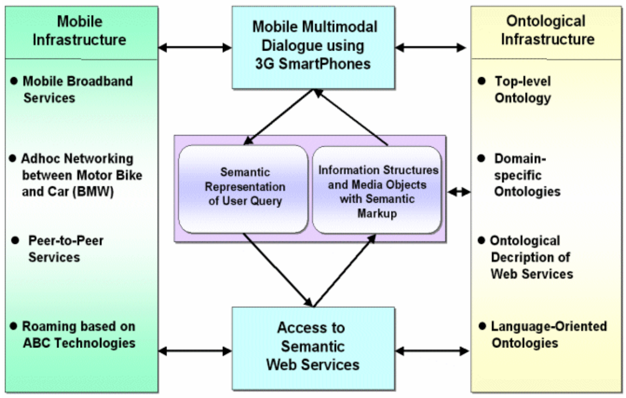
4. ON- AND OFFLINE EXTRACTION OF SEMANTIC STRUCTURES

Extraction from syntactic web pages deals with the development and application of techniques that render answers to arbitrary, domain-independent user requests online and in real-time. These methods are rather shallow but fast and robust. They depend on already existing search engines like Google and only take classic ("syntactic") web pages into account that are written in pure HTML (without semantic annotation).

The Offline-Extraction component SOBA consists of a web crawler, linguistic annotation components and a component for the transformation of linguistic annotations into an ontology-based representation which can be used to answer user queries. Currently, SOBA is applied to FIFA web pages to extract information related to the world cup 2006.

5. WEB SERVICES

The SmartWeb system utilizes existing web services, including the T-Info web services. They include navigational and weather information. For use within the SmartWeb system, the web services are semantically annotated. Certain queries invoke calls to corresponding web services whose responses can be fed back to the user.



6. DEMONSTRATION

We will demonstrate the SmartWeb system on the PDA, which accepts input in spoken form, by keyboard, or by pen. Users can ask open questions which are linguistically analysed and passed on to the semantic mediator subsystem. The semantic mediator queries all available knowledge sources, including the manually created domain ontologies, semantically annotated web services, the knowledge which was automatically extracted from web pages, and a linguistic query-answering subsystem on the open internet.

Answers returned by the knowledge sources come accompanied by multimedia data and are endowed with certainty estimates. They are integrated in order to obtain the system response, consisting of text and multimedia objects. Further queries by the user are then processed in the context of the previous interactions.

Metaphor-based Semantic Browsing in M-FIRE

Andrea Proli
DEIS, University of Bologna
Bologna, Italy

Stefano Rizzi
DEIS, University of Bologna
Bologna, Italy

Matteo Golfarelli
DEIS, University of Bologna
Bologna, Italy

Keywords

Semantic Web, RDF, Visual Interfaces

1. INTRODUCTION AND MOTIVATION

Though the syntax of RDF is designed to be human-readable, most end-users are not familiar with it. Thus, tools have been developed that (1) generate visual presentations of RDF statements and (2) translate user actions performed on those presentations into queries over the RDF knowledge base. An open problem in this field is to guarantee a satisfactory compromise between expressivity and domain-independence. The former is meant as the capability of delivering an intuitive visualization of knowledge and some tailored navigation primitives to end-users working in a given application domain, while the latter is aimed at accomplishing a high degree of reusability. Most existing tools, e.g. [2, 3], favor domain-independence by visually presenting constructs – such as classes and specializations – that are familiar to knowledge engineers but not to domain experts. The same holds for most Protégé plug-ins except Jambalaya [1], that allows users to associate custom semantics to the same graphical primitive, namely containment. Indeed, though domain-specific formalisms have a lower degree of reusability, they provide graphically richer constructs better understood by domain experts.

An approach to achieve a nice trade-off between reusability and expressivity is to decouple the mechanism for transforming RDF documents into an expressive visualization from the criteria that drive the transformation. In this demonstration we present M-FIRE (Metaphor-based Framework for Information Representation and Exploration), a configurable framework for semantic browsing of RDF-based knowledge, relying on the adoption of custom *metaphors*. Metaphors drive the process through which visual presentations are obtained for a given document, and define how queries are generated upon user actions. M-FIRE generalizes the approach pursued by current tools, which provide representations for *individuals* only, by allowing metaphors to specify the representation of more complex information patterns, namely *sets of statements*. The demonstration will be focused on (1) showing how users can perform semantic browsing by relying on domain-specific and intuitive visualizations of concepts, thus interacting in a simple manner with complex knowledge, and (2) illustrating how flexibility and reusability are effectively achieved in our framework by the use of metaphors.

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006),* Budva, Montenegro, 11th - 14th June, 2006

2. APPROACH OVERVIEW

The overall functional architecture of our framework is sketched in Figure 1. First of all, we rely on an *RDF server* that, given a SPARQL query, returns a result as an RDF document (from now on, the *source*). We will use the Jena engine to this end. The *metaphor selector* takes the source and returns the best suited metaphor for its fruition, according to the vocabulary of the source. The metaphor consists of a *representation metaphor* and a *navigation metaphor*. The representation metaphor is given as input to the *representer*, which applies the directives contained in it to generate a *representation* describing in an abstract form, independently of any implementation detail, how concepts will be visualized. Then, a properly chosen *encoder* translates the representation into a concrete form, called *encoding* (e.g., an HTML document), which can be given as input to the end-user's rendering program (e.g., a Web browser). The choice of the best suited encoder for a given representation is carried out by the *encoder selector*, again based on the representation vocabulary.

Once rendering has been completed by the rendering program, end-users are allowed to interact with it. Events generated by user actions are captured by the *controller*, which creates an *event description* in the form of an OWL document describing the occurred event (for instance, a user's double click on an icon representing a soccer player). The event description is then given as input to the *navigator* together with the chosen navigation metaphor. In the same way as the representation metaphor tells the representer which representation must be produced for a given source, the navigation metaphor tells the navigator which SPARQL query must be formulated for a given event. The resulting query is then forwarded to the RDF server, and the process is repeated.

Some details on the different phases are given in the following subsections.

2.1 Representation

Representation is the process of obtaining a document in which certain graphical drawings are associated to certain (kinds of) statements belonging to the source, according to the directives contained in the representation metaphor. In order to provide a general solution, representation is split into two phases, namely *enrichment* and *mapping*, and the representation metaphor is split accordingly, namely into an *enrichment metaphor* and a *mapping metaphor*.

Enrichment exploits the enrichment metaphor to augment the source with new classifications and concept definitions. This is done by launching the Pellet reasoner (which sup-

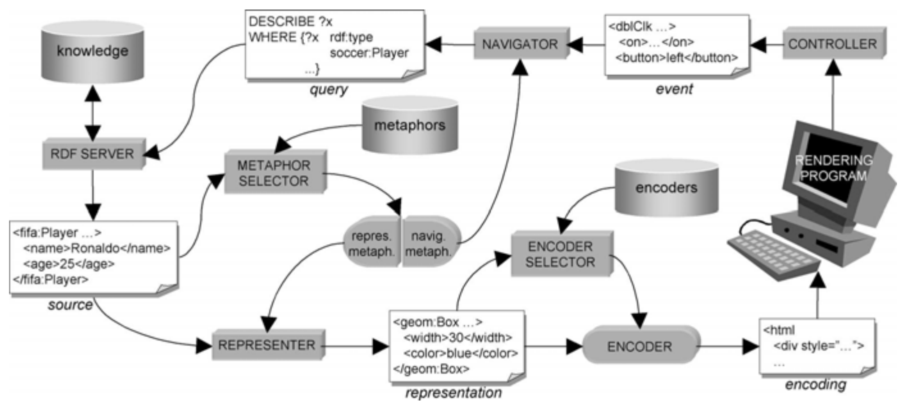


Figure 1: Overall functional architecture for M-FIRE

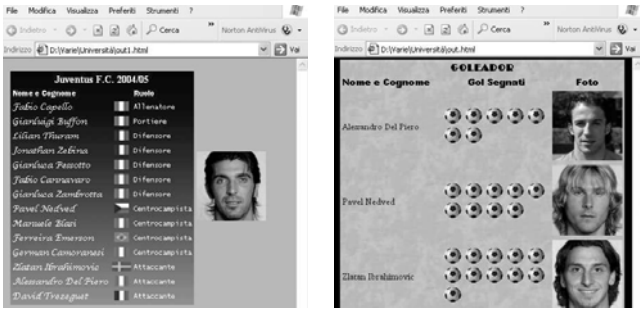


Figure 2: Two representations obtained by applying different metaphors to the same source

ports OWL-DL reasoning) on the merge between the source and the enrichment metaphor. Mapping interprets the directives contained into the mapping metaphor to carry out the association of particular visual items and graphical styles to certain kinds of statements in the document resulting from enrichment. This is done by translating the mapping metaphor into a set of SPARQL `CONSTRUCT` queries, executed by the ARQ SPARQL engine, whose results are then properly merged to create the representation. Notably, a crisp separation between enrichment and mapping gives metaphor designers a better control over the kind of inferences and classifications that are performed, also increasing the flexibility and the modularity of design.

2.2 Encoding

Encoding consists in translating the representation into a document that can be parsed by a proper program to produce a graphical rendering. Many formats could be used to this end: for instance, two circles connected by a line could be encoded as both a GraphML document and an SVG document; a table containing names and photos could be encoded as an SVG document as well as an HTML document. We developed two different encoders, one for HTML and one for GraphML. Figure 2 shows the HTML rendering of two representations obtained by applying different representation metaphors to the same RDFS source containing information about soccer players. To the left, soccer teams are the focus of interest and they are rendered as a list of players; to the right, goalscorers are shown together with their score.

A relevant issue concerning encoding is *semantic annotation*, that traces a correspondence between the graphical items used to represent a given set of statements and the represented statements themselves. Such correspondence is first established at a conceptual level by the representer, then embedded into the encoding. There, such annotations can be used by the end-user's rendering program to integrate graphical information with semantic information.

2.3 Navigation

Navigation is the interaction schema triggered by a user action upon the visual presentation of a given piece of knowledge, and it consists in translating that action into a query over the underlying knowledge base. In order to enhance flexibility, the directives for this translation are contained in a *navigation metaphor*. Remarkably, the representation and the navigation metaphors are independent of each other: e.g., a navigation metaphor could state that a double click on an item representing a soccer player should trigger a query for retrieving the soccer team in which that player is enrolled, whatever its visual representation is.

Navigation is carried out in two steps. First, the *controller* captures the actions performed by the user on the current rendering and describes them in the form of an OWL document, by relying on the semantic annotations in the encoding. The controller needs to be tightly integrated with the rendering program; we developed two controllers: one is a simple HTML viewer that uses the Microsoft WebBrowser ActiveX control for rendering, the other is a plug-in for Protégé that uses the yFiles library for rendering GraphML documents. Then, the *navigator* parses a set of directives contained in the navigation metaphor to produce, from the event description, a `DESCRIBE` SPARQL query which is then sent to the RDF server for execution.

3. REFERENCES

[1] M. Storey et al. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé. In *Proc. Workshop on Interactive Tools for Knowledge Capture*, 2001.

[2] F. Van Harmelen et al. Ontology-based information visualisation. In *Proc. Workshop on Visualization of the Semantic Web*, pages 546–554, 2001.

[3] R. Volz et al. KAON SERVER - a semantic web management system. In *Proc. WWW*, 2003.

An Authorization Scenario for S-OGSA

Pinar Alper, Oscar Corcho, Michael Parkin, Ioannis Kotsiopoulos, Paolo Missier, Sean Bechhofer, Carole Goble
School of Computer Science. University of Manchester
Oxford Road. M13 9PL. Manchester. United Kingdom
+44 (0)161 2756821
<penpecip, ocorcho, parkinm, ioannis, pmissier, seanb, carole>@cs.man.ac.uk

ABSTRACT

The Semantic Grid initiative aims to exploit knowledge in the Grid to increase the automation, interoperability and flexibility of Grid middleware and applications. To bring a principled approach to developing Semantic Grid Systems, and to outline their core capabilities and behaviors, we have devised a reference Semantic Grid Architecture called S-OGSA. We present the implementation of an S-OGSA observant semantically-enabled Grid authorization scenario, which demonstrates two aspects: 1) the roles of different middleware components, be them semantic or non-semantic, and 2) the utility of explicit semantics for undertaking an essential activity in the Grid: resource access control.

Keywords

Semantic Grid, architecture, authorization, S-OGSA.

1. Grid and Semantic Grid

The Grid vision is defined as the next generation infrastructure that will enable coordinated, well-controlled sharing of resources through dynamic, transient confederations known as Virtual Organizations (VOs). The roadmap for the realization of this vision is elaborated in the Open Grid Services Architecture (OGSA) [3]. The OGSA view of the Grid is comprised of a 3-tiered service-oriented architecture, where applications are brought together with Grid resources through a layer of middleware services. In the middleware layer OGSA defines certain service categories as core capabilities that Grids should have: Security, Resource Management, Execution Management, Optimization, Data, Information and Self-Management.

The Semantic Grid initiative aims to foster the progress in the realization of the Grid vision by extending it so that resource metadata is exposed and handled explicitly, and shared and managed via Grid protocols. To date, the application of Semantic technologies to the grid has been through exploratory experimentation where pioneering applications combining Grid and Semantic technologies were built.

In order to provide a systematic approach to building Semantic Grid systems and to outline their architectural organization and interaction patterns we have developed S-OGSA [1], a reference Semantic Grid architecture. S-OGSA extends the core capability set of OGSA by adding another category of services: the **Semantic Provisioning Services**. This category is decomposed at least into 4 sub-categories, namely Ontology Services, Metadata Services, Annotation Services and Reasoning services. All together these services are responsible for generating, managing and exploiting semantically-encoded metadata in the Grid. Furthermore, S-OGSA defines the **Semantically Aware Grid Services** as middleware services that provide an OGSA enumerated capability but differ from others by being capable of operating over explicit semantics.

2. A Scenario in the Insurance Domain

We have implemented a role based access control system for an International Insurance Settlement Grid. The scenario requires that Customers should be allowed to make Insurance Policy Applications based on an evaluation of their previous car insurance and accident history. Close investigation of this scenario has revealed that it can be cast as a Grid authorization scenario. Prior to detailing the implementation we will briefly cover the background technologies that have been used in it.

2.1 Background on Authorization

Authorization falls in the scope of the Security category in OGSA. It is normally needed after the authentication of a client, so as to decide whether or not it can access a specific resource. The OGSA-AuthZ framework[i] describes different authorization models, architectures, components and systems that are currently used to support authorization in Grid applications.

Authorization decisions are based on the information available from the client and on the list of rules in a particular expression language that govern whether or not access requests will be approved, namely the authorization or access control policies. Among the languages used to represent authorization request/response messages and access control policies the most complete is XACML[ii].

Our implementation conforms to the OGSA-AuthZ framework and uses XACML to deliver request/response messages.

2.2 Declarative Approach to Authorization

Access control policies can be expressed in different ways and with different languages, and are usually distributed among the organizations belonging to a VO, so that we can talk about central and local policies. One common example of an access control policy is an access control list, which may control the access to

Demos and Posters of the 3rd European Semantic Web conference (ESWC 2006), Budva, Montenegro, 11th-14th June, 2006

specific resources from individual users as well as from the groups they belong to and/or from the roles they play in the VO.

Access control lists and similar specifications are useful and work in many contexts, but they may not be sufficient when it comes to expressing more complex access control policies or when users, groups or roles cannot be easily expressed by enumeration, due to the existence of a large number of users or to the dynamicity of the user base. This is the situation in our insurance case study: we cannot pre-determine the eligibility of each customer for insurance application at the time the access control policies are made. We can, however, specify access control rules based on the roles that a customer plays. These roles are defined in terms of certain restrictions on the customer properties, and are obtained at run-time taking into account the customer's properties.

To define complex roles declaratively we have decided to use and extend the KAoS suite of ontologies[iii]. This ontology set contains descriptions about actors, groups, actions, resources, policy types, etc., and are extended with concepts related to the insurance domain (accidents, insurance companies, customers, etc.). Furthermore, we define customer roles that will be used to express the access control policies. Examples or such roles are GoodReputationDriver (a driver whose accident record contains at most one claim and who has been registered with an insurance company), BadReputationDriver (a driver whose accident record shows three or more claims), etc.

3. System Architecture

Figure 1 shows the component interactions: *1)* a Grid-enabled **Ontology Access Service**, WS-DAIOnt [2], responsible for hosting and managing the VO ontologies, *2)* a set of **Metadata Services,** powered by the Atlas P2P RDF storage and querying system [4], which store insurance customer metadata (Policy Information Point-PIP), *3)* the **Reasoning Service**, which is a description logic classifier used to infer customer roles based on their properties, *4)* the (XACML compliant) **Authorization Service**, which evaluates the access control function in the system (Policy Decision Point-PDP) and 5) the **CarFraud Service**, through which customers make their insurance policy applications (Policy Enforcement Point-PEP).

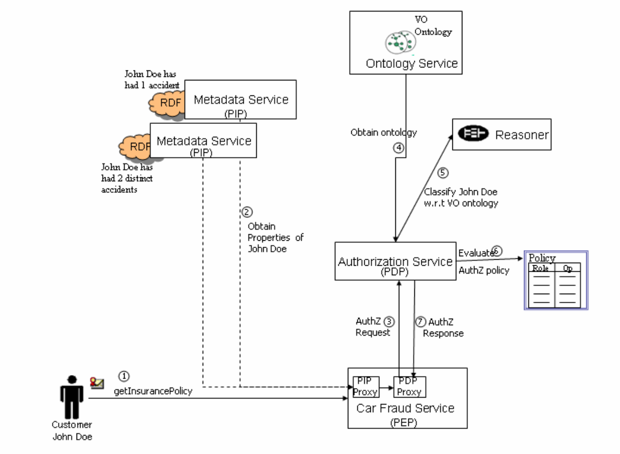


Figure 1. Authorisation scenario architecture.

The system operates as follows: An Insurance Customer makes a policy application by calling the associated method of the CarFraud Service (Step 1). The Car Fraud service delegates the eligibility evaluation of this person to the local authorization proxies. These proxies first contact the Metadata Service to obtain the properties of the customer (Step 2). Once customer metadata is gathered, an XACML Authorization request is generated for the Subject with an attribute containing the RDF based metadata regarding this subject (Step 3). Upon receiving the request the Authorization service contacts the ontology service to obtain the VO ontology containing the role definitions (Step 4). The ontology together with the customer metadata is passed onto the Reasoner to infer the role of the insurance customer (Step 5). Once the customer's roles are inferred the Authorization services evaluates the access control function using this information (Step 6) and returns a Permit/Deny/Indeterminate result to the Car Fraud Service's authorization proxy (Step 7).

4. Conclusions

With our implementation we have aimed to demonstrate:

The Semantic Grid Ecosystem of Services. The scenario demonstrates how a Semantically Aware Grid Service, namely the Authorization Service, uses some S-OGSA Semantic Provisioning Services, namely Metadata, Ontology and Reasoning, to deliver enhanced functionality via exploiting semantic metadata.

Grid Compliant Semantic Middleware. The services in the scenario are WS-RFiv compliant Grid services running on the Globus Toolkit 4v container. We believe it is important for the Semantic technologies and tools to be Grid enabled so as to enable their uptake by the Grid community.

Flexibility of Declarative Approaches for Authorization. The role-based authorization mechanism is based on dynamically inferring customer roles using a reasoner over OWL concept descriptions and instance data.

5. Acknowledgements

This work has been done as part of the EU IST Project OntoGrid (FP6-511513) and of the EU Marie Curie Fellowship RSSGRID (FP6-2002-Mobility-5-006668).

6. REFERENCES

[1] Corcho et al. An Overview of S-OGSA: a Reference Architecture for the Semantic Grid. Journal of Web Semantics. 2006. To appear

[2] Esteban et al. WS-DAIOnt: Ontology Access Provisioning in Grid Environments. GGF16 Semantic Grid workshop. 2006.

[3] Foster et al. The Open Grid Services Architecture Version 1.0. Specification, GGF OGSA Working Group, 2005.

[4] Koubarakis et al. Semantic Grid Resource Discovery using DHTs in Atlas. GGF16 Semantic Grid workshop. 2006.

---

[i] https://forge.gridforum.org/projects/ogsa-authz/
[ii] http://www.oasis-open.org/committees/xacml
[iii] http://ontology.ihmc.us/kaos.html
[iv] http://www.oasis-open.org/committees/wsrf
[v] http://www.globus.org/toolkit

M-OntoMat-Annotizer: Linking Ontologies with Multimedia Low-Level Features for Automatic Image Annotation

Carsten Saathoff, Norman Timmermann, Steffen Staab
ISWeb
University of Koblenz, D-56016 Koblenz
{saathoff,normania,staab}@uni-koblenz.de

Kosmas Petridis, Dionysios Anastasopoulos, Yiannis Kompatsiaris
Informatics and Telematics Institute
GR-57001 Thermi-Thessaloniki, Greece
{kosmas,anastas,ikom}@iti.gr

1. INTRODUCTION

Understanding and semantic annotation of multimedia content have been identified as important steps towards more efficient manipulation and retrieval of visual media. Although new multimedia standards, such as MPEG-7, provide important functionalities for manipulation and transmission of objects and associated metadata, the extraction of semantic descriptions and annotation of the content with the corresponding metadata is out of the scope of these standards and is still left to the content manager. However, since the generation of annotations manually is tedious and often expensive, methods to automatically annotate images with semantic descriptions are under active research. In the aceMedia IST FP6 project[1], ontologies and Semantic Web technologies are employed in order to achieve a knowledge-based semantic analysis of multimedia content. This allows for more generic algorithms not limited to specific objects, but capable to handle a diverse number of concepts depending on the provided domain knowledge.

In this paper we present M-OntoMat-Annotizer, a tool covering the step of knowledge acquisition for automatic annotation of multimedia content. The tool allows to extract MPEG-7 visual descriptors [1] from both images and videos and to store these descriptors as so-called visual prototypes of ontology classes. The prototypes are stored as RDF instances using a RDF version of the MPEG-7 visual descriptors. The prototype approach specifically provides an OWL-DL friendly way of linking classes to concrete visual characteristics. In the following we first present the overall knowledge-assisted analysis framework, and then continue with M-OntoMat-Annotizer. We conclude with a short outlook on future work.

2. KNOWLEDGE ASSISTED ANALYSIS

Analysis of multimedia content is under active research for several years now, and progress has been made in several application domains, such as person detection or face recognition. However, all these approaches have in common that they are limited to very specific problems and lack the applicability for other domains. In [2], we propose an architecture for automatic annotation of multimedia content that is independent of specific algorithms, but uses ontologies enriched with low-level features to label regions in images with semantic concepts.

In order to handle the semantic gap in multimedia content interpretation, aceMedia proposed and implemented a comprehensive ontology infrastructure. An important part of this infrastructure is the Visual Descriptor Ontology (VDO), developed to link ontology concepts to low-level visual descriptors. It is based on MPEG-7 but modeled in RDFS, which allows for the direct integration with other RDF data used throughout the project. The descriptors are represented as so called *prototypes*, which are instances of the domain concepts linked to specific visual descriptors. The additional super-concept *Prototype* assures that prototypical instances can later be distinguished from the "real" metadata. By using the prototype approach to represent the visual features of concepts, we avoid direct linking of concepts to instances, and the ontologies are kept OWL DL compatible. Details about the aceMedia Knowledge Infrastructure and the VDO in particular can be found in [3].

We will shortly outline the analysis procedure for still images. Initially the image is segmented into a number of regions. For each region the MPEG-7 visual descriptors are extracted and then compared to the prototype instances stored in the active domain ontology. Using this approach, for each domain concept a distance to the descriptors of the region can be computed. This allows to decide which concept provides the best match for the specific region. Finally, the region is labeled with the concept providing the smallest distance. Apparently, the algorithm is domain independent, since it uses a generic distance computation which only relies on the visual descriptors. The concepts that can be detected, and especially the definition of the concepts, are completely defined in the ontologies and the extracted visual prototypes, so that switching the algorithm to another domain could be easily achieved by providing a different domain ontology and according prototypes.

[1] `http://www.acemedia.org/`

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

3. M-ONTOMAT-ANNOTIZER

In order to exploit the ontology infrastructure mentioned above and enrich the domain ontologies with multimedia descriptors, M-OntoMat-Annotizer (M stands for Multimedia) [3] was implemented. The development was based on an extension of the CREAM (CREAting Metadata for the Semantic Web) framework [4] and its reference implementation OntoMat-Annotizer[2].

For this reason, the Visual Descriptor Extraction (VDE) tool was implemented as a plug-in to OntoMat-Annotizer and is the core component for supporting the initialization of RDF domain ontologies with low-level multimedia features. The VDE plug-in manages the overall low-level feature extraction and linking process by communicating with the other OntoMat-Annotizer components.

The VDE Visual Editor and Media Viewer presents a graphical interface for loading and processing of visual content (images and videos), visual feature extraction and linking with domain ontology concepts. The interface, as shown in Fig. 1, seamlessly integrates with the common OntoMat-Annotizer ones. Usually, the user needs to extract the visual features (i.e. descriptors included in the VDO) of a specific object inside the image/frame. M-OntoMat-Annotizer lets the user draw a region of interest in the image/frame and apply the multimedia descriptor extraction only to the specific selected region. Alternatively, M-OntoMat-Annotizer also supports automatic segmentation of the image/frame; whenever a new image/frame is loaded it is automatically segmented into regions. The user can then select a desired region or even merge two or more regions and proceed with the extraction.

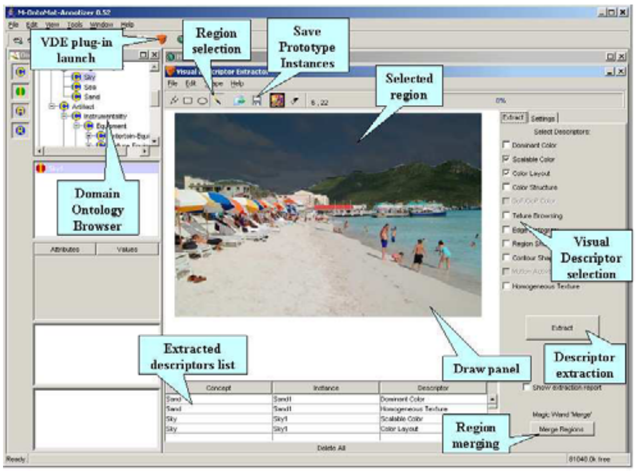


Figure 1: The M-OntoMat-Annotizer user interface

By specifying an instance of a concept in the ontology browser and selecting a region of interest, the user can extract and link appropriate visual descriptor instances with instances of domain concepts that serve as *prototypes* for these concepts. The created statements are added to the knowledge base and can be retrieved in a flexible way during multimedia content analysis. M-OntoMat-Annotizer saves the domain concept prototype instances together with the corresponding descriptors, in a separate RDFS file and leaves the original domain ontology unmodified.

M-OntoMat-Annotizer is publicly available as free software through the aceMedia web site since last May[3]. An updated version of the tool is expected to be published during summer 2006.

4. CONCLUSIONS

In this paper we presented M-OntoMat-Annotizer, a tool for enriching domain ontologies with MPEG-7 visual descriptors expressed in RDF. We also presented an approach to exploit the stored information for the automatic and domain independent annotation of images.

We currently plan further extensions of the tool. The main focus will be on the implementation of a high-level multimedia annotation tool based on M-OntoMat-Annotizer. Apparently this denotes the opposite direction of its current purpose, i.e. not annotating the ontologies with low-level features, but annotating the content with semantic metadata. Obviously, using the current plug-in, annotations could be made on a region level. Especially using the automatic segmentation capability of M-OntoMat-Annotizer, the detailed annotation would become less tedious. Furthermore, the generation of such annotations leads to the second planned extension: the extraction of spatial, topological and contextual knowledge from annotated content that can be used for multimedia reasoning and improve the automatic annotation significantly. Therefore, the tool can both be used by users to annotate their images for later retrieval or organization, but also as a means to generate a-priori knowledge useful for the knowledge-assisted analysis of multimedia content and multimedia reasoning.

Acknowledgments This research was partially supported by the European Commission under contract FP6-001765 aceMedia. The expressed content is the view of the authors but not necessarily the view of the aceMedia project as a whole.

5. REFERENCES

[1] ISO/IEC 15938-3 FCD Information Technology - Multimedia Content Description Interface - Part 3: Visual, March 2001, Singapore.

[2] T. Athanasiadis, V. Tzouvaras, K. Petridis, F. Precioso, Y. Avrithis and I. Kompatsiaris. Using a Multimedia Ontology Infrastructure for Semantic Annotation of Multimedia Content. In *Proceedings of the 5th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2005) at the 4th International Semantic Web Conference (ISWC 2005)*, Nov. 2005.

[3] S. Bloehdorn, K. Petridis, C. Saathoff, N. Simou, V. Tzouvaras, Y. Avrithis, S. Handschuh, I. Kompatsiaris, S. Staab, and M.G. Strintzis. Semantic Annotation of Images and Videos for Multimedia Analysis. In *Proceedings of the 2nd European Semantic Web Conference (ESWC 2005)*, May 2005.

[4] Siegfried Handschuh and Steffen Staab. Cream - creating metadata for the semantic web. *Computer Networks*, 42:579–598, AUG 2003. Elsevier.

[2] `http://annotation.semanticweb.org/ontomat/`

[3] `http://www.acemedia.org/aceMedia/results/software/m-ontomat-annotizer.html`

Mapping Attempto Controlled English to OWL DL

Kaarel Kaljurand
Institute for Computational Linguistics
University of Zurich
kalju@ifi.unizh.ch

Norbert E. Fuchs
Institute for Computational Linguistics
University of Zurich
fuchs@ifi.unizh.ch

ABSTRACT

We describe ongoing work on the mapping between Attempto Controlled English (ACE) and OWL DL. ACE is a well-studied controlled language, with a parser that converts ACE texts into Discourse Representation Structure (DRS). We describe a relatively direct mapping of a subset of the DRS language to OWL DL. This mapping renders ACE an interesting companion to existing OWL front-ends.

1. INTRODUCTION

Existing OWL tools (e.g. Protégé, SWOOP, SemanticWorks) are user-friendly graphical point-and-click editors but for complex class descriptions they require the user to possess a large knowledge of Description Logics. E.g. [3] list the problems that users encounter when working with OWL and express the need for a "pedantic but explicit" paraphrase language.

We envision a text based system that allows the users to express the ontologies in the most natural way — in natural language. Such a system would be easy to use since it does not presuppose a knowledge of mathematical concepts such as disjointness or transitivity. The system would be tightly integrated with an OWL reasoner, but the output of the reasoner (if expressed in OWL as a modification of the ontology) would again be verbalized in natural language, so that all user interaction takes place in natural language.

As a basis of the natural language, we have chosen Attempto Controlled English (ACE), a subset of English that can be converted through its DRS representation into first-order logic representation and automatically reasoned about (see [1] for more information). The current version of ACE offers language constructs like countable and mass nouns, collective and distributive plurals, generalized quantifiers, indefinite pronouns, noun phrase/verb phrase/sentence negation, and anaphoric references to noun phrases through proper names, definite noun phrases, pronouns, and variables. The intention behind ACE is to minimize the number of syntax and interpretation rules needed to predict the resulting DRS, or for the end-user, the reasoning results. The small number of ACE function words have a clear and predictable meaning and the remaining content words are classified only as verbs, nouns, adjectives and adverbs. Still, ACE has a relatively complex syntax compared to the OWL representation e.g. in the OWL Abstract Syntax specification, but as ACE is based on English, its grammar rules are intuitive i.e. already known to English speakers.

Some existing results show the potential and the need for a natural language based interface to OWL. [2] paraphrase OWL class hierarchies but their target is not a controlled language and cannot be edited and parsed back into a standard OWL representation. [4] propose writing ontologies in a controlled language, but do not provide a natural syntax for writing TBoxes. In the following, we describe a mapping from a subset of ACE (OWL ACE) to OWL DL (in RDF/XML notation) and conclude with an overview of the remaining work.[1]

2. FROM ACE TO OWL

Figure 1 shows an ACE text and its corresponding DRS that makes use of a small number of predicates, most importantly `object` derived from nouns and `predicate` derived from verbs. The predicates share information by means of discourse referents (denoted by capital letters) and are further grouped by embedded DRS boxes, that represent implication (derived from *if... then...* or *every*), negation (derived from various forms of English negation), and disjunction (derived from *or*). Conjunction — derived from relative clauses, explicit *and*, or the sentence end symbol — is represented by the co-occurrence in the same DRS-box.

The mapping to OWL does not modify the existing DRS construction algorithm but only the interpretation of the DRS. It considers everything in the toplevel DRS to denote individuals or relations between them. Individuals are introduced by nouns, so that propernames map to individuals with type *owl:Thing* and common nouns to an anonymous individual with the type derived from the corresponding noun (e.g. class *Man*). Properties are derived from

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

[1] A demo of this mapping is available among the Attempto tools at `http://www.ifi.unizh.ch/attempto/tools`

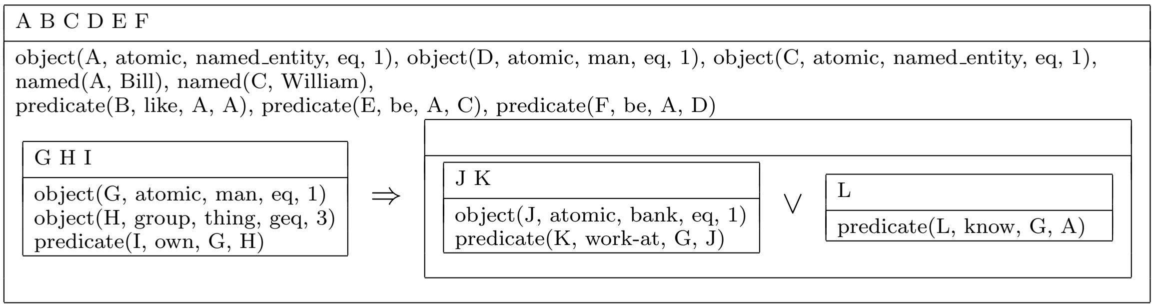


Figure 1: DRS corresponding to the ACE text "Bill who is a man likes himself. Bill is William. Every man who owns at least 3 things works-at a bank or knows Bill." Note that the DRS has been simplified for layout purposes. Also, the example is somewhat artificial to demonstrate concisely the features of OWL as expressed in ACE.

transitive verbs. A special meaning is assigned to the copula 'be' which introduces an identity between individuals. An embedded implication-box introduces a *subClassOf* relation between classes: the head of the implication maps to the subclass description, the body to its superclass description. Transitive verbs introduce a property restriction with *someValuesFrom* a class denoted by the object of the verb, and the copula introduces a class restriction. Negation and disjunction boxes in the implication-box introduce *complementOf* and *unionOf*, respectively. Any embedding of them is allowed. The plural form of the word 'thing' which can be modified by a number allows to define cardinality restrictions. Thus the DRS of figure 1 has the following meaning (in Description Logics notation):

$$\text{bill} \in \top,\ \text{m1} \in \text{Man},\ \text{william} \in \top,$$
$$\text{bill} = \text{m1},\ \text{bill} = \text{william},$$
$$\text{likes(bill, bill)}$$

$$\text{Man} \sqcap \text{owns} \geq 3 \sqsubseteq$$
$$\exists\ \text{worksAt Bank} \sqcup \exists\ \text{knows}\ \{\text{bill}\}$$

ACE can also describe OWL properties (super property, inverse property and transitivity) but this sounds quite "mathematical", e.g. transitivity is expressed as "If a thing A is taller than a thing B and B is taller than a thing C then A is taller than C.". On the other hand there does not seem to be a better way in natural languages.

Note that the mapping does not target all the syntactic variety defined in the OWL specification, e.g. elements like *disjointWith* or *equivalentProperty* cannot be directly expressed in ACE, but their semantically equivalent constructs can be generated.

Given that ACE is easy to learn and use, can we say the same about OWL ACE? With regards to full ACE, OWL ACE introduces a number of restrictions: there is no support for ditransitive and intransitive verbs, prepositional phrases, adverbs, intransitive adjectives and most forms of plurals. Furthermore, there are restrictions to the DRS structure which are more difficult to explain to the average user, e.g. disjunction is not allowed to occur at the toplevel DRS ("John sees Mary or John sees Bill."). A further restriction could require the predicates in the implication-box to share one common discourse referent as the subject argument, and not to share the object arguments. This would allow us to exclude sentences like "If a man sees a mouse then a woman does not see the mouse." which does not seem to map nicely to an ontology language but instead to a rule language. Then again, this restriction is too strong as it would exclude property expressions ("Everybody who loves somebody likes him/her.") and a way to express *allValuesFrom* ("Everything that a herbivore eats is a plant.").

3. FUTURE WORK

The current mapping lacks support for datatype properties and enumerations (*oneOf*). Furthermore, there is only a limited support for *someValuesFrom* and *allValuesFrom*, meaning that not all the possible configurations of these constructs can be generated with ACE. We will add support of those constructs along with support of URIs for naming classes, properties and individuals.

We will also implement the mapping from OWL to ACE which must handle all OWL constructs, some of which the ACE-to-OWL mapping does not produce. The mapping from OWL to ACE must also deal with the naming conventions of OWL constructs.

4. REFERENCES

[1] Attempto project. Attempto website, 2006. `http://www.ifi.unizh.ch/attempto`.

[2] Daniel Hewlett, Aditya Kalyanpur, Vladamir Kovlovski, and Chris Halaschek-Wiener. Effective Natural Language Paraphrasing of Ontologies on the Semantic Web. In *End User Semantic Web Interaction Workshop (ISWC 2005)*, 2005.

[3] Alan L. Rector, Nick Drummond, Matthew Horridge, Jeremy Rogers, Holger Knublauch, Robert Stevens, Hai Wang, and Chris Wroe. OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns. In *EKAW 2004*, 2004.

[4] Rolf Schwitter. Controlled Natural Language as Interface Language to the Semantic Web. In *2nd Indian International Conference on Artificial Intelligence (IICAI-05)*, Pune, India, December 20–22nd 2005.

System for semi-automatic ontology construction

Blaž Fortuna
Blaz.Fortuna@ijs.si
Institute Jožef Stefan
Jamova 39
1000, Ljubljana

Marko Grobelnik
Marko.Grobelnik@ijs.si
Institute Jožef Stefan
Jamova 39
1000, Ljubljana

Dunja Mladenič
Dunja.Mladenic@ijs.si
Institute Jožef Stefan
Jamova 39
1000, Ljubljana

ABSTRACT

In this paper, we review two techniques for topic discovery in collections of text documents (Latent Semantic Indexing and K-Means clustering) and present how we integrated them into a system for semiautomatic topic ontology construction. The system offers supports to the user during the construction process by suggesting topics and analyzing them in real time.

General Terms

Algorithms, Human Factors.

Keywords

Semi-automatic Ontology construction, Background knowledge

1. INTRODUCTION

When working with large corpora of documents it is hard to comprehend and process all the information contained in them. Standard text mining and information retrieval techniques usually rely on word matching and do not take into account the structure of the documents within the corpus. We try to overcome that by automatically extracting the topics covered within the documents from the corpus and helping the user to organize them into a topic ontology.

Topic ontology is a set of topics connected with different types of relations. Each topic includes a set of related documents. Construction of such ontology from a given corpus can be a very time consuming task for the user. In order to get a feeling on what the topics in the corpus are, what the relations between topics are and to assign each document to some certain topics, the user has to go through all the documents and process them manually. We tried to overcome this by building *OntoGen*, a special tool which helps the user by suggesting the possible new topics and visualizing the topic ontology created so far, all in real time. OntoGen, in combination with the corpus visualization tools [4], aims at assisting the user in a fast semi-automatic construction of the topic ontology from a large document collection.

This paper is organized as follows. In Section 2 we present text mining techniques that are used in OntoGen, and in Section 3 we give a short demonstration of the tool and its features.

2. TEXT MINING TECHNIQUES

2.1 Representation of text documents

In order to use the algorithms we will describe later we must first represent text documents as vectors. We use standard Bag-of-Words (BOW) approach together with the TFIDF weighting [5]. This representation is often referred to as vector-space model. The similarity between two documents is defined as the cosine of the angle between their vector representations – cosine similarity.

2.2 Latent Semantic Indexing

The language contains much redundant information, since many words share common or similar meaning. For computer this can be difficult to handle without some additional information (background knowledge). Latent Semantic Indexing (LSI), [3], is a technique for extracting this background knowledge from text documents. It uses a technique from linear algebra called Singular Value Decomposition (SVD) and bag-of-words representation of text documents for detecting words with similar meanings. This can also be viewed as extraction of hidden semantic concepts or topics from the text documents.

2.3 K-Means clustering

Clustering is a technique for partitioning data so that each partition (or cluster) contains only points which are similar according to some predefined metric. In the case of text this can be seen as finding groups of similar documents, that is documents which share similar words.

K-Means [6] is an iterative algorithm which partitions the data into k clusters. It has already been successfully used on text documents [7] to cluster a large document corpus based on the document topic.

2.4 Keywords extraction

We used two methods for extracting keywords from a given set of documents: (1) keyword extraction using centroid vectors and (2) keyword extraction using Support Vector Machine (SVM) [2]. We used this two methods to generate description for a given topic based on the documents inside the topic.

The first method works by using the centroid vector of the topic (centroid is the sum of all the vectors of the document inside the topic). The main keywords are selected to be the words with the highest weights in the centroid vector. The second method is based on the idea presented in [1] which uses SVM binary classifier. Let A be the topic which we want to describe with keywords. We take all the documents from the topics that have A for a subtopic and mark these documents as negative. We take all the documents from the topic A and mark them as positive. If one document is assigned both negative and positive label we say it is positive. Then we learn a linear SVM classifiers on these documents and classify the centroid of the topic A. Keywords describing the concept A are the words, which's weights in SVM normal vector contribute most when deciding if centroid is positive.

The difference between these two approaches is that the second approach takes into account the context of the topic. Let's say that we have a topic named 'computers'. When deciding, what the keywords for some subtopic A are, the first method would only look at what the most important words within the subtopic A are and words like 'computer' would most probably be found important. However, we already know that A is a subtopic of 'computers' and we are more interested in finding the keywords that separate it from the other documents within the 'computers' topic. The second method does that by taking the documents from all the super-topics of A as a context and learns the most crucial words using SVM.

3. SEMI-AUTOMATIC CONSTRUCTION OF TOPIC ONTOLOGY

We view semi-automatic topic ontology construction as a process where the user is taking all the decisions while the computer only gives suggestions for the topics, helps by automatically assigning documents to the topics, helps by suggesting names for the topics, etc. The suggestions are applied only when the users decides so. The computer also helps by visualizing the topic ontology and the documents.

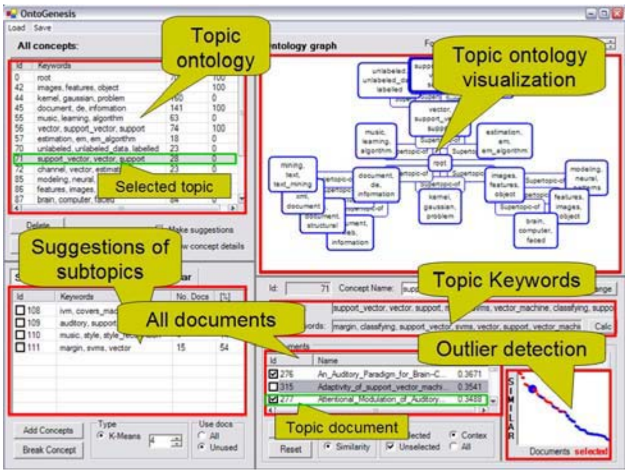


Figure 1. Screen shot of the interactive system OntoGen for construction of topic ontologies.

In Figure 1 you can see the main window of the interactive system we developed. The system has three major parts that will be further discussed in following subsections. In the central part of the main window is a visualization of the current topic ontology (Ontology visualization). On the left side of the window is a list of all the topics from this ontology. Here the user can select the topic he wants to edit or further expand into subtopics. Further down is the list of suggested subtopics for the selected topic (Topic suggestion) and the list with all topics that are in relation-ship with the selected topic. At the bottom side of the window is the place where the user can fine-tune the selected topic (Topic management).

3.1 Ontology visualization

While the user is constructing/changing topic ontology, the system visualizes it in real time as a graph with topics as nodes and relations between topics as edges. See Figure 1 for an example of the visualization.

3.2 Topic suggestion

When the user selects a topic, the system automatically suggests what the subtopics of the selected topic could be. This is done by LSI or k-means algorithms applied only to the documents from the selected topic. The number of suggested topics is supervised by the user. Then, the user selects the subtopics he finds reasonable and the system automatically adds them to the ontology with relation 'subtopic-of' to the selected topic. User can also decide to replace the selected topic with the suggested subtopics. In Figure 1 you can see how is this feature implemented in our system.

3.3 Topic management

The user can manually edit each of the topics he added to the topic ontology. He can change which documents are assigned to this topic (one document can belong to more topics), what is the name of the topic and what is the relationship of the topic to other topics. The main relationship is subtopic-of and is automatically added when adding subtopics as described in the previous section. The user can control all the relations between topics by adding, removing, directing and naming the relations.

Here the system can provide help on more levels:

- The system automatically assigns the documents to a topic when it is added to the ontology.
- The system helps by providing the keywords describing the topic using the methods described in Section 3. This can assist user when naming the topic.
- The system computes the cosine similarity between each document from the corpus and the centroid of the topic. This information can assist the user when searching for documents related to the topic. The similarity is shown on the list of documents next to the document name and the graph of similarities is plotted next to the list. This can be very practical when searching for outliers inside the concepts or for the documents that are not in the concepts but should be in considering their content.

REFERENCES

[1] Brank J., Grobelnik M., Milic-Frayling N. & Mladenic D. Feature selection using support vector machines. Proc. of the Third International Conference on Data Mining Methods and Databases for Engineering, Finance, and Other Fields, 2002.

[2] Cristianini N. & Shawe-Taylor, J., An introduction to support vector machines, Cambridge University Press

[3] Deerwester S., Dumais S., Furnas G., Landuer T. & Harshman R. Indexing by Latent Semantic Analysis, J. of the American Society of Information Science, vol. 41/6, 391-407

[4] B. Fortuna, D.Mladenic, M. Grobelnik. Visualization of text document corpus. Informatica journal 29 (2005), 497-502

[5] Grobelnik M. & Mladenic D. Automated knowledge discovery in advanced knowledge management. J. of. Knowledge management 2005, Vol. 9, 132-149.

[6] Jain, A. K., Murty M. N., & Flynn P. J. Data Clustering: A Review, ACM Computing Surveys, vol 31/3, 264-323, 1999.

[7] Steinbach, M., Karypis, G., Kumar, V. A comparison of document clustering techniques. In Proceedings of KDD Workshop on Text Mining, pp. 109110, 2000

Oyster – Sharing and Re-using Ontologies in a Peer-to-Peer Community

Raul Palma
Facultad de Informática. Universidad Politécnica de Madrid
Campus Montegancedo, s/n 28860
Boadilla del Monte. Madrid. Spain
34-913367439
rpalma@fi.upm.es

Peter Haase
Institute AIFB
Universität Karlsruhe (TH)
D-76128 Karlsruhe, Germany
49-7216083705
haase@aifb.uni-karlsruhe.de

Asunción Gómez-Pérez
Facultad de Informática. Universidad Politécnica de Madrid
Campus Montegancedo, s/n 28860
Boadilla del Monte. Madrid. Spain
34-913367439
asun@fi.upm.es

ABSTRACT

In this poster, we present Oyster, a Peer-to-Peer system for exchanging ontology metadata among communities in the Semantic Web. Oyster exploits semantic web techniques in data representation, query formulation and query result presentation to provide an online solution for sharing ontologies, thus assisting researchers in re-using existing ontologies.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Information networks*
K.6.4 [**System Management**]: Centralization/decentralization

General Terms

Management, documentation, design, reliability

Keywords

Ontology, Peer-to-Peer, Repository, Metadata

1. INTRODUCTION

Currently efficient knowledge sharing and reuse is rather difficult, as it is hard to find and share ontologies available among the community due to the lack of standards for documenting and annotating ontologies with metadata information. This raises the problem of having many isolated ontologies created by many different parties. Besides the costs of the duplicate efforts, this also hampers interoperability between ontology-based applications. Oyster[1] is a Peer-to-Peer application that exploits semantic web techniques in order to provide a solution for exchanging and re-using ontologies. To achieve this, Oyster implements a proposal for a metadata standard, called Ontology Metadata Vocabulary (OMV)[2] [2] which is based on discussions and agreements carried out in the EU IST thematic network of excellence Knowledge Web[3] as a way to describe ontologies. The decentralized approach provides an ideal solution for users that require a repository to which they have full access and can perform any operation without any consequences to other users. For example, users from academia or industry might use a personal repository for a task dependent investigation, or ontology engineers, might use it during their ontology development process to capture information about different ontology versions. We argue that a decentralized system is the technique of choice, since it allows the maximum of individuality while it still ensures exchange with other users. A centralized approach, on the other hand, allows reflecting long-term community processes in which some ontologies become well accepted for a domain or community and others become less important. However, both approaches could be combined to cover a variety of use cases.

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva Montenegro, 11th - 14th June, 2006

2. OYSTER

Oyster provides an innovative solution for sharing and re-using knowledge (i.e. ontologies), which is a crucial step to enable Semantic Web. The Oyster system has been implemented as an instance of the Swapster system architecture[4]. In Oyster, ontologies are used extensively in order to provide its main functions (importing data, formulating queries, routing queries, and processing answers).

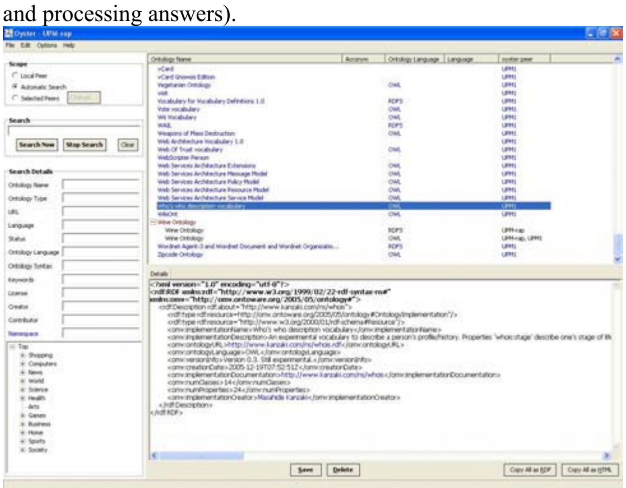


Figure 1. Oyster Screenshot

Creating and Importing Metadata: Oyster enables users to create metadata about ontologies manually and also to import ontology files in order to automatically extract the ontology metadata available and let the user to fill in the missing values. For the automatic extraction, Oyster supports the OWL[5], DAML+OIL[6], and RDF-S[7] ontology languages. The ontology metadata entries are aligned and formally represented according to two ontologies: (1) the proposal for a metadata standard OMV

[1] Available at http://oyster.ontoware.org/
[2] More information at http://omv.ontoware.org/
[3] http://knowledgeweb.semanticweb.org/

[4] http://swap.semanticweb.org/
[5] http://www.w3.org/TR/owl-guide/
[6] http://www.w3.org/TR/daml+oil-reference
[7] http://www.w3.org/TR/rdf-schema

that describes the properties of the ontology, and (2) a topic hierarchy (i.e. DMOZ[8]) that describes specific categories of subjects to define the domain of the ontology.

Formulating Queries: Users can search the repository for ontologies by means of simple keyword searches, or more advanced, semantic searches (c.f. the left pane of figure 1). Queries are formulated in terms of these two ontologies. This means that queries can refer to fields like name, acronym, ontology language, etc. or they may refer to topic terms.

Routing Queries: As shown in the upper left pane of figure 1, users may query a single specific peer (e.g. their own computer, or a certain peer because this peer is known as a big provider of information), or they may query a specific set of peers (e.g. all the members of a specific organization), or they may query the entire network of peers (e.g. when the user has no idea where to search). In the latter case, queries are routed automatically through the network depending on the expertise of the peers, describing which topic of the topic hierarchy a peer is knowledgeable about. In order to achieve this expertise based routing, a matching function determines how closely the semantic content of a query matches the expertise of a peer [1].

Processing results: The results matching a query are presented in a result list (c.f. upper right pane in figure 1). The answer of a query might be very large and may contain many duplicates due to the distributed nature and potentially large size of the Peer-to-Peer network. Such duplicates might not be exact copies because of the semi structured nature of the metadata, so the ontologies are used again to measure the semantic similarity between different answers and to remove apparent duplicates. Then a merged representation that combines the knowledge from the individual and potentially incomplete items is presented to the user. Details of the particular results are shown in the lower right side of Figure 1. Users can save the results of a query into their local repository for future use.

3. OMV IN OYSTER

Oyster provides an infrastructure for storing, sharing and finding ontologies making use of the proposal for a metadata standard OMV. OMV comprises the OMV Core, which captures information relevant to most of the ontology reuse settings and various OMV Extensions that allow ontology developers/users to specify task/application-specific ontology-related information (e.g. ontology merging, alignment or versioning, evaluation of ontologies or ontological engineering methodologies). These extensions should be compatible to the OMV core, but at the same time fulfill the requirements of a domain, task or community-driven setting. The OMV elements are classified according to the type and purpose of the contained information such as availability (e.g. URI, URL), provenance (e.g. creator, contributor), applicability (e.g. domain, ontology type), relationship (e.g. import, backward compatibility), format (e.g. ontology language and syntax), statistics (e.g. number of classes or properties) and general information (e.g. name, description). Furthermore, OMV classifies elements according to their impact on the prospected reusability of the described ontology content as required, optional, and extensional. OMV also models additional classes and properties required to support the reuse of ontologies, especially in the context of the Semantic Web, such as Party, Organisation, Person, LicenseModel, OntologyLanguage, OntologySyntax and OntologyTask,. For a complete description of OMV please refer to [2].

[8] http://dmoz.org/

4. RELATED WORK

A closely related application is the **Onthology**[9] central repository, which also exploits the OMV. Onthology offers a complementary application to Oyster as both applications have a different usage perspective: Oyster as a decentralized system is the technique of choice for users who needs the maximum of individuality while still ensuring exchange with other users with up-to-date information. Ontology as a centralized system allows reflecting long-term community processes in which some ontologies become well accepted for a domain or community and others become less important. There exists similar approaches to our proposed solution, but in general their scope is quite limited. E.g. the **DAML ontology library**[10] provides a catalog of DAML ontologies that can be browsed by different properties. The **FIPA ontology service**[11] defines an agent wrapper of open knowledge base connectivity. The Semantic Web search engine **SWOOGLE**[12] makes use of particularly metadata which can be extracted automatically. Finally the **SchemaWeb Directory**[13] is a repository for RDF schemas expressed in RDFS, OWL and DAML+OIL.

5. CONCLUSIONS AND FUTURE WORK

To conclude, the reuse of existing ontologies within communities is a key issue for sharing knowledge on the Semantic Web. This task, however, is rather difficult because of the heterogeneity, distribution and diverse ownership of the ontologies as well as the lack of sufficient metadata. As we summarized in this paper, our contribution, Oyster, addresses exactly these challenges by implementing a proposed standard for metadata for describing ontologies. Oyster is already being applied in the KnowledgeWeb project which has partners across the European Union. Oyster is ranked as the number one in the list of top downloaded projects of Ontoware[14] (650 downloads, including all versions and releases). Currently, there are around 250 ontologies shared in Oyster network. We are in the process of collecting usage statistics. Finally, our future work includes addressing many challenges like the integration of Oyster with central repository, handle change propagation, evaluation of expertise ranking, using trust information and evaluation of performance.

6. ACKNOWLEDGMENTS

Our thanks to our partners from the EU projects Knowledge Web (FP6-507482) and NeOn (FP6-27595) for their present and future collaboration.

7. REFERENCES

[1] Haase P. et al. Peer selection in peer-to-peer networks with semantic topologies. In proc of ICNSW'04, Paris, June 2004.

[2] Hartmann, J., Palma, R.OMV-Ontology Metadata Vocabulary for the semantic web.2005. http://omv.ontoware.org

[9] http://www.onthology.org/
[10] http://www.daml.org/ontologies
[11] http://www.fipa.org/specs/fipa00086/XC00086C.html
[12] http://swoogle.umbc.edu/
[13] http://www.schemaweb.info/
[14] http://ontoware.org/

Storing and Querying RDF Data in Atlas*

Zoi Kaoudi, Iris Miliaraki, Matoula Magiridou, Antonios Papadakis-Pesaresi
and Manolis Koubarakis

Dept. of Informatics and Telecommunications
National and Kapodistrian University of Athens
Panepistimioupolis, Ilissia
Athens 15784 Greece
{zoi,iris,matoula,apapadak,koubarak}@di.uoa.gr

Keywords
RDF, query processing, Semantic Web, peer-to-peer networks, DHT, scalability.

1. INTRODUCTION
In recent years, more and more resources are semantically annotated, thus generating huge amounts of RDF metadata. Current centralized RDF repositories lack the required scalability and fault tolerance to deal with this emerging situation. Therefore, the need for a scalable system that will be able to scale to millions of RDF triples is becoming prevalent. Distributed hash tables (DHTs) is a recent P2P technology that has been proposed for the scalable and fault-tolerant storage and querying of RDF data [2, 1]. Since annotation is by itself a distributed process it ties very well with the model of work imposed by P2P systems.

In this demo paper, we present Atlas, a P2P system for the distributed storage and retrieval of RDF data. Atlas is built on top of the distributed hash table Bamboo[1] and supports pull and push querying scenarios. It inherits all the nice features of Bamboo (openness, scalability, fault-tolerance, resistance to high churn rates) [10] and extends Bamboo's protocols for storing and querying RDF data. In the OntoGrid project, Atlas is used to implement the metadata service of S-OGSA, a new architecture for the Semantic Grid [3].

2. ATLAS ARCHITECTURE
Nodes in an Atlas network are organized in an identifier ring using the Bamboo DHT protocol. Nodes can enter RDF data into the network and pose RQL [6] queries. In the typical Atlas application that we envision, RDF data will be used to describe resources owned by network nodes (e.g., ontologies or services). Atlas supports two querying scenarios: *one-time* querying and *publish/subscribe*. Each time a node poses an one-time query, the network nodes cooperate to find RDF data that form the answer to the query. In the publish/subscribe scenario, a node subscribes with a continuous query. A continuous query is indexed somewhere in the network and each time matching RDF data is published, Atlas nodes cooperate to notify the subscriber. A high level view of the Atlas architecture is shown in Figure 1.

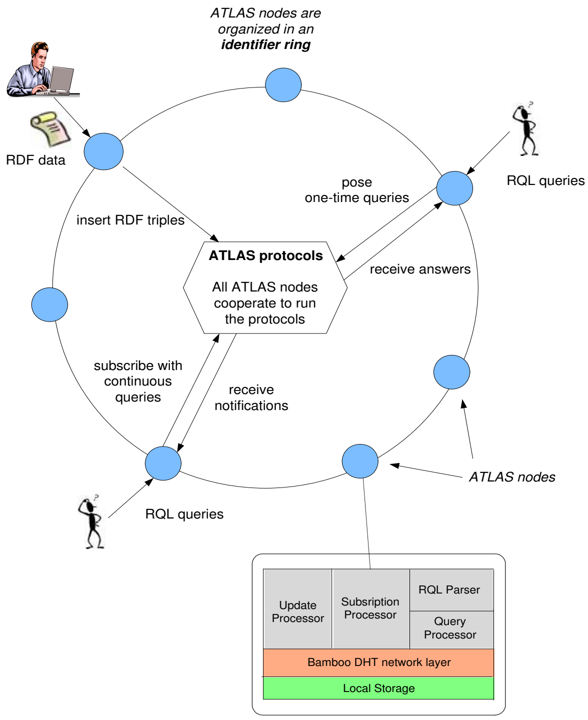


Figure 1: Atlas architecture

In the following, we describe the architecture of each node, which is similar to the architecture of [2] (see Figure 1). We distinguish between six components in an Atlas node:

*This work is partially funded by FP6/IST project OntoGrid. The work was performed while the authors were with the Dept. of Electronic and Computer Engineering, Technical University of Crete.

[1] `http://bamboo-dht.org/`

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

the *update processor*, the *subscription processor*, the *query processor*, the *RQL parser*, the *local storage* and the *Bamboo DHT network layer*, which is responsible for routing and handling the network messages. We will describe briefly the role and functionality of each component.

The update processor is the component that stores an RDF document or a bunch of RDF data in the Atlas network. It takes as input an RDF document in RDF/XML or N3 format, it decomposes the document into triples and stores the triples in the network. Each triple is stored in the local storage of the appropriate node based on an identifier of the triple. We have used the triple indexing algorithm presented in [2]. The update processor is also responsible for updating or removing triples previously stored in the network. Updates in Atlas are not supported yet but will eventually follow the semantics of RUL [9].

The subscription processor is the component providing a publish/subscribe functionality in Atlas. This component is under development using the ideas presented in [8], which significantly extend the publish/subscribe algorithms of [2].

The query processor is the component responsible for evaluating the queries. The query processing algorithm of Atlas deals with *conjunctive triple-pattern* queries, an extension of the class of queries considered in [2]. The protocols followed for the evaluation of queries are described in [5].

The RQL parser is the component which takes as input RQL queries posed by the node. It parses the query and if it is correctly formed, it produces a conjunctive triple-pattern query and passes it to the query processor. Currently, Atlas supports data RQL queries expressed in RQL.

The local storage is the place where each node stores locally its $(key, value)$ pairs. In the Bamboo implementation, the Berkeley DB database [4] is used. Berkeley DB is an open source database library that provides a simple API for data access and management.

3. DEMONSTRATION SCENARIO

Let us now describe a complete demonstration scenario of our system. Initially, an Atlas network is created by a single node that bootstraps by itself. Then, other nodes join the network following the join protocol described in [5]. The application scenario demonstrated is based on the well-known problem of resource discovery [7]. Two actors are involved in this scenario, the resource providers and the resource consumers.

Resource providers want to publish their resources and create resource descriptions expressed using the RDF data model. They store them in the Atlas network using a store operation. Resource consumers want to discover resources that meet their needs. They pose appropriate RQL queries to search for resource information stored in Atlas.

4. CONCLUSION

We presented Atlas, a P2P system for the distributed storage and retrieval of RDF data. Atlas is implemented on top of the Bamboo DHT network and provides clients with the ability to store and query RDF data. In the future, we plan to expand the functionality of the Atlas system by supporting queries for RDFS and the update language RUL.

5. REFERENCES

[1] Karl Aberer, Philippe Cudre-Mauroux, Manfred Hauswirth, and Tim Van Pelt. GridVine: Building Internet-Scale Semantic Overlay Networks. In *Proceedings of the Thirteenth International World Wide Web Conference (WWW2004)*, New York, May 2004.

[2] Min Cai, Martin R. Frank, Baoshi Yan, and Robert M. MacGregor. A Subscribable Peer-to-Peer RDF Repository for Distributed Metadata Management. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 2(2):109–130, December 2004.

[3] Carole Goble, Ioannis Kotsiopoulos, Oscar Corcho, Paolo Missier, Pinar ALPER, and Sean Bechhofer. S-OGSA as a Reference Architecture for OntoGrid and for the Semantic Grid. In *GGF16 Semantic Grid Workshop*, February 2006.

[4] BerkeleyDB: http://sleepycat.com/.

[5] Zoi Kaoudi, Iris Miliaraki, Spiros Skiadopoulos, Matoula Magiridou, Erietta Liarou, Stratos Idreos, and Manolis Koubarakis. Specification and Design of Ontology Services and Semantic Grid Services on top of Self-organized P2P Networks. Deliverable D4.1, Ontogrid project, September 2005.

[6] Greg Karvounarakis, Sofia Alexaki, Vassilis Christophides, Dimitris Plexousakis, and Michel Scholl. RQL: A Declarative Query Language for RDF. In *Proceedings of the 11th International World Wide Web Conference*, May 2002.

[7] Manolis Koubarakis, Zoi Kaoudi, Iris Miliaraki, Matoula Magiridou, and Antonios Papadakis-Pesaresi. Semantic Grid Resource Discovery using DHTs in Atlas. In *GGF16 Semantic Grid Workshop*, February 2006.

[8] Erietta Liarou, Stratos Idreos, and Manolis Koubarakis. Publish-Subscribe with RDF Data over Large Structured Overlay Networks. In *Proceedings of the 3rd International Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P 2005)*, Trondheim, Norway, 28-29 August.

[9] Matoula Magiridou, Stavros Sahtouris, Vassilis Christophides, and Manolis Koubarakis. RUL: A Declarative Update Language for RDF. In *Proceedings of the Fourth International Semantic Web Conference (ISWC2005)*, 2005.

[10] Sean Rhea, Dennis Geels, Timothy Roscoe, and John Kubiatowicz. Handling Churn in a DHT. In *USENIX Annual Technical Conference*, 2004.

Ontology-Based Legal Information Retrieval to Improve the Information Access in e-Government

Asunción Gómez-Pérez
Facultad de Informática. Universidad Politécnica de Madrid
Campus Montegancedo, s/n 28860 Boadilla del Monte. Madrid. Spain
34-913367439
asun@fi.upm.es

Fernando Ortiz-Rodríguez
Facultad de Informática. Universidad Politécnica de Madrid
Campus Montegancedo, s/n 28860 Boadilla del Monte. Madrid. Spain
34-913363670
fortiz@fi.upm.es

Boris Villazón-Terrazas
Facultad de Informática. Universidad Politécnica de Madrid
Campus Montegancedo, s/n 28860 Boadilla del Monte. Madrid. Spain
34-913363670
bvillazon@fi.upm.es

ABSTRACT
In this paper, we present EgoIR, an approach for retrieving legal information based on ontologies; this approach has been developed with Legal Ontologies to be deployed within the e-government context.

Categories and Subject Descriptors
H.3.3 Information Storage and Retrieval: Information Search and Retrieval – *query formulation, retrieval models, search process*.

General Terms
Design, Experimentation

Keywords
Ontology, Information Retrieval

1. INTRODUCTION
For more than two decades, the AI and Law community has been very active and productive. In the early 80´s, research was focused on logic programming. Other approach adopted was the case-based reasoning. Knowledge Engineering was also of interest for the research community and the field most applied since it allowed developing and using the legal ontologies that underlie the growth of the Semantic Web.

The e-Gov has been strengthened with all these previous studies carried out by the research community and now its main concern is data representation and information management. By its nature, the e-Gov is supported by the legal domain.

Our contribution consists of an ontology based approach for legal information retrieval that we called EgoIR. This system has as a main goal to retrieve e-Gov documentation. EgoIR deals with Real-estate transaction documents, and gives an opportunity to the citizens, business and governments to integrate and recover documents. For this purpose EgoIR provides facilities for managing, searching and sharing e-Gov documentation.

2. EgoIR
EgoIR is an Ontology-Based Legal Information Retrieval System. This system is the result of integrating Ontological Workbench WebODE[1], and a text search engine library, Lucene[2]. In this section we describe the system architecture and the Legal Ontologies.

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 11th - 14th June, 2006

2.1 Architecture
The system integration of the EgoIR is built and composed by the Search Client, the Search Server and the Ontology Server modules, which are described in the next subsections. Figure 1 shows the general architecture of the system.

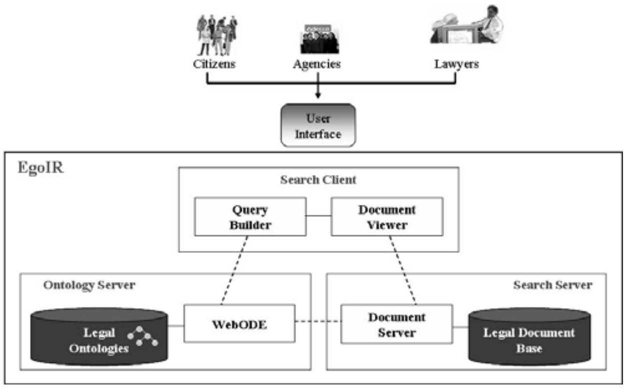


Figure 1. EgoIR System Architecture.

2.1.1 Ontology Server
This module defines how the knowledge is structured in the application domain. This module includes the Legal Ontologies within WebODE [2].

Within the Legal Ontologies, concept instances are associated with documents. Every time that a new concept instance is added the Ontology Server communicates with the Search Server to index its corresponding document.

2.1.2 Search Client
This module incorporates two sub-modules: a Query Builder and a Document Viewer.

Query Builder connects to the Ontology Server, in order to access Legal Ontologies, browse them and obtain concepts to build the query by using a graphical interface. This module sends the query to the Search Server.

[1] http://webode.dia.fi.upm.es/

[2] http://jakarta.apache.org/lucene

Document Viewer connects to the Search Server, in order to retrieve the legal documents satisfying the query, and to the Ontology Server to browse and display the documents.

This module is also a procedural mapping module that makes possible the interoperability between WebODE and Lucene. Figure 2 shows the relationships between ontology and Lucene index.

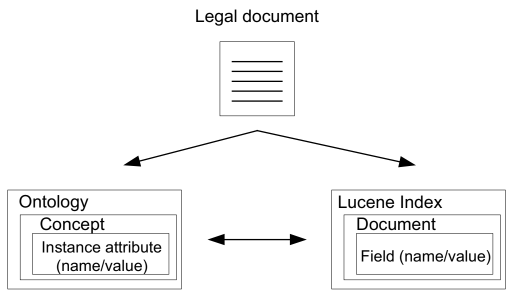


Figure 2. Relationships between Ontology and Lucene Index

2.1.3 Search Server

The Search Server module is based on Lucene and processes the Legal Document Base to create internally access structures. These structures (called indices) allow fast document location and are stored locally in the file system of the operating system.

The Legal Document Base consists of electronic documents that are stored in the file system. These electronic documents are: juridical term glossary, models of contracts, legal norms and jurisprudence. Currently the document's annotation process is manually done. When a concept instance is created, using WebODE interface, the values from its instance attributes are indexed using Lucene which includes the electronic document.

This module also performs a kind of mapping between ontology instances and indexed documents. We can see in Figure 3 that the query concepts with their syno-nyms are sent to the Search Server and it searches on the Lucene index, more specifi-cally inside Legal Document that represents a Generic Document Concept. The Legal Document has inside the content field that represents an instance attribute that holds the document's content.

2.2 Legal Ontologies

Legal Ontologies [3] were built to represent the real-state transactions in the Spanish Government domain. These Legal Ontologies were developed with knowledge acquired by experts from academic and private sectors and built with the methodology METHONTOLOGY [2] and the workbench WebODE [2].

For the EgoIR sytem eleven ontologies have been developed: *person*, *civil personality*, *organization*, *location*, *tax*, *contract model*, *jurisprudence*, *Real-estate transaction verifications*, *Real-estate*, *legislation*, and *Real-estate transaction*.

3. RELATED WORK

There are many systems developed for managing legal information, but only a few deals with legal knowledge. In this section we describe briefly some legal IR systems.

In [5], CLIME (Computerized Legal Information Management and Explanation) aims at improving the access and understanding of large collections of legal information through the Internet. CLIME just combines conventional IR with artificial legal reasoning without ontologies. In [1], the authors describe the Webocrat system whose goal is to provide new types of communication and service flows from public institutions toward citizens, thus improving the access of citizens to services and information of public administration. This system focuses on security issues.

Another work reported in [4] is the EULEGIS (European User Views to Legislative Information in Structured Form), whose main goal is to provide a consistent user interface for legal IR generated in different legal systems and at different legislative levels. This system focuses on user interfaces.

4. CONCLUSIONS AND FUTURE WORK

In this paper we present our first approach to an ontology-based legal IR, which aims to retrieve government documents in a timely and accurate way. This is an approach of an entirely new wave of legal knowledge systems. At this time we can mention that the utility of ontologies within an IR is twofold: On the one hand, as a social impact, ontologies are a good way to guide user to the legal terms, thus avoiding him/her to make mistakes at the query construction; and on the other hand, mostly technical, ontologies are a key to the development the Semantic Web and improving interoperability on the legal applications.

Finally, in the near future we will improve the performance of EgoIR and we will focus on further enhancement of the ontology-based retrieval mechanism by means of Natural Language Processing (NLP) techniques for an user friendlier environment; on the automatic semantic annotation of the documents to improve the search process; and on security issues by providing a summary of the retrieved documents.

5. ACKNOWLEDGMENTS

We would like to thanks to Rosario Plaza and Ángel López-Cima. This work is carried out within the ongoing Reimdoc Project which is supported by the Spanish Technology and Science Ministry (Project FIT-340100-2004-22).

This project is partially funded by a scholarship granted through the Program of Teaching Staff Improvement (PROMEP) at the Tamaulipas University.

6. REFERENCES

[1] Dridi, F., Pernul, G.: The Webocracy project: Overview and Security Aspects. In Schnurr et al., ProffessionellesWissensmanagement: Erfahrungen und Visionen. Shaker Verlag, Aachen, 2001.

[2] Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering. Springer Verlag. 2003

[3] Gómez-Pérez, A., Ortiz-Rodríguez, F., Villazón-Terrazas, B.: Legal Ontologies for the Spanish e-Government. In Proceedings of the 11th Conference of the Spanish Association for Artificial Intelligence. Santiago de Compostela, Spain, 2005.

[4] Lyytikinen, V., Tiitinen, P., Salminen, A.: Challenges for European Legal Information Retrieval. Proceedings of the IFIP 8.5 Working Conference on Advances in Electronic Government. Zaragoza, Spain, 2000.

[5] Winkels, R.G.F.: CLIME: Legal Information Serving Put to the Test. Pre-proceedings of the Second French-American Conference on AI and LAW. Available online at: http://www.Iri.jur.uva.nI/~winkels/papers/AIL98-paper.html

HyperCuP Lightweight Implementation

A Universal Solution for Making Application Distributed

Sławomir Grzonkowski
WETI, Gdansk University of Technology, Poland
DERI, NUI Galway, Ireland
slawomir.grzonkowski@deri.org

Sebastian Ryszard Kruk
DERI, NUI Galway
IDA Business Park
Lower Dangan, Galway, Ireland
sebastian.kruk@deri.org

ABSTRACT

Contemporary applications need an efficient solution for communication to implement robust information retrieval mechanisms and fault tolerant networks. Apart from implementing an robust, scalable communication protocol the solution should be accessible with easy to use API that would not require too much of an effort to use it.

In this article we present HyperCuP Lightweight Implementation (HLI) which delivers an alternative P2P architecture based on web services. This implementation has already been deployed with diverse systems like JeromeDL, a semantic digital library and FOAFRealm, a distributed identity management system based on social networking.

We describe an architecture of the HyperCuP Lightweight Implementation. We show how to deploy it with one's own application and how to take advantage of the established hypercube topology.

Keywords

Distributed Computing, HyperCuP, P2P networks

1. INTRODUCTION

The contemporary applications must be able to process many queries per second, especially digital libraries that are affiliated with large universities and host huge databases to thousands of students. Because of the fact that universities keep both daily and extra-mural studies, digital libraries are overloaded during end-of-term examinations period. Furthermore, many of these libraries offer fancy features like collaborative groups, searching in network of federated libraries or Single sing-on registration. Unfortunately, as long as digital libraries do not utilize semantics, users will repeat similar queries many times, because first search results usually do not respond to desired information being sought.

Operations like looking for resources and authentication in distributed environment cause undesirable network traffic. Our work identifies and combines several techniques from the Semantic Web to P2P networks, which results in improving efficiency of communication in e.g. searching for resources and managing users profiles.

The remainder of this short article is organized as follows: section 2 describes problems and requirements of distributed systems. Section 3 provides a short description of the HyperCuP Lightweight Application. Finally in section 4 we describe the overview of the demo we would like to present during ESWC 2006.

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006
.

2. P2P INFRASTRUCTURE FOR SCALABLE DISTRIBUTED COMMUNICATION

Eventhough most of contemporary applications implement distributed (or sometimes even ubiquitous) computing paradigm there is lack of support for developing this paradigm in a lightweight fashion. Although the requirements are usually similar, we can found as many various solutions as projects we encounter. Unfortunately, hardly any of existing solution have satisfied our requirements. First of all, the application like a digital library needs an efficient broadcast algorithm. Moreover, during the search process all nodes must be equally balanced in order to prevent from Denial Of Service (DoS) attack. Secondly, new digital library servers should not affect the overall network efficiency. Therefore the solution has to be scalable. Finally, we required an open-source lightweight framework that could be easily adapted to existing applications delivering new axis of distributed computing with least effort possible.

After investigating the problem we have encountered the idea of HyperCuP (Hyper Cube in P2P) network. The HyperCuP [4] protocol was invented by Schlosser, Sintek, Decker and Nejdl as a P2P protocol based on a topology also known as Caley graph structure.

The protocol provides a fast and an efficient broadcast algorithm which sends the minimum number of messages across the network. Moreover, HyperCuP lets nodes to join and leave the network at any time. The HyperCuP infrastructure tends to be balanced most of the time. This can help in prevent the application utilizing HyperCuP for communication from Distributed Denial of Service attacks. In the balanced stated, a total number of messages sent to the network in each broadcast is always equal to `log(n)`, where n is the number of nodes in the network.

The reference implementation of HyperCuP has been developed in the Edutella [1] project. Although the source code of Edutella is available as an opensource project, we could not extract the actual core of the HyperCuP protocol to use it in our projects. In addition, Edutella contains many modules which are firmly depended each other. Those facts induced us to design and implement our own application. Based on the requirements presented earlier we have decided to take the lightweight approach.

3. HYPERCUP LIGHTWEIGHT IMPLEMENTATION

The aim of HyperCuP Lightweight Implementation (HLI) implementation is to make the opensource system that provides an easy to use, lightweight framework for extending almost any kind of applications with distributed computing features. HyperCuP provides programmer friendly API that do not require too much effort in order to start using it in existing projects. This section provides a short overview of the architecture and describes the practical aspects of using HLI.

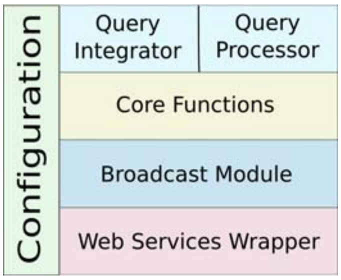


Figure 1: Architecture diagram

3.1 Architecture

The overall architecture of HyperCuP Lightweight Implementation consists of five modules (see Fig. 1). The *Web Services Integration* module is responsible for communication between different instances of HLI. This module is responsible for working in distributed heterogeneous environments. Web services support allows significant interoperability in HLI and delivers the process of turning on SSL support is far easier.

The *Broadcast Module* allows user to decide how a peer (an instance of HLI-enabled application) behaves upon the arrival of the request from the network. According to the lightweight approach only a couple of requirements have to be met to enable HyperCuP in the application. The most important one is to implement the `LocalQuery` interface (located in the *Query Processor* module) by the external application. The interface has only one method `performQuery` which is invoked when the broadcast message arrives to the peer. The implementation of this method changes the actual behavior of the peer.

The *Broadcast Query Integrator* module delivers the implementation of the broadcast processing that is independent on the actual application that is HLI-enabled. The final results of the broadcast message consists of the request from the sender and responses from the all peers along the paths integrated in this peer.

Finally the *Core Functions* module delivers HyperCuP protocol essential code for creating networks, joining peers or monitoring the state of the network.

It is worth to mention that the implementation details are transparent from the user–programmer perspective. HyperCuP Lightweight Implementation required implementation of only one one interface in order to make application work. Additionally managing behaviour of HLI can be done with *Configuration* module.

3.2 Practical Use

Deploying Lightweight HyperCuP Implementation with existing application requires several steps. In the beginning, the HyperCuP component has to be initialized by setting some attributes like the address of the web services interface of this HyperCuP component and the implementation of the local query interface (currently defined in Java API). Performing the query results in invoking implementation of the method `performQuery` that should be registered during the initialization step. In result the query is being executed on every node of the P2P network along the broadcast paths. There are no constraints on either an implementation of the `performQuery` method or the way the query message should be handled, except the requirement that objects passed as parameters to this method have to implement the Java `Serializable` interface.

One additional step is required when running application for the first time. Since the topology has to be set up, peers must connect to the HyperCuP network by connecting to any peer in the network. According to the HyperCuP protocol this the connection request is routed to the appropriate peer in order to keep the network in a balanced state.

4. PRESENTATION PLAN

During the demo session we will present how to deliver distributed paradigm with HLI in a couple of steps to an existing application. The demonstration will consist of:

1. Deployment with an example web-application.
2. Preparation and implementation of an example query.
3. Connecting nodes to the hypercube topology.
4. Executing the query and show the results.
5. Presentation of existing solutions based on HyperCuP:
    - *distributed search protocol* in JeromeDL [3] - a semantic digital library.
    - *distributed authentication protocol* in D-FOAF [2] a distributed identity management system.

4.1 Acknowledgments

This work was supported by Science Foundation Ireland Grant No. SFI/02/CE1/I131 and by the Knowledge Web project (FP6 - 507482) and partially by KBN, Poland under grant No. 4T11C00525. The authors would like to acknowledge Stefan Decker, Paweł Bugalski, the DERI Semantic Web Cluster and the Corrib.org working group for fruitful discussions.

5. REFERENCES

[1] Edutella project: http://edutella.jxta.org/.
[2] S. Grzonkowski, A. Gzella, H. Krawczyk, S. R. Kruk, F. J. M.-R. Moyano, and T. Woroniecki. D-FOAF - Security Aspects in Distributed User Managment System. In *TEHOSS'2005*.
[3] S. R. Kruk, S. Decker, and L. Zieborak. JeromeDL - Adding Semantic Web Technologies to Digital Libraries. In *DEXA Conference*, 2005.
[4] M. Schlosser, M. Sintek, S. Decker, and W. Nejdl. Ontology-Based Search and Broadcast in HyperCuP. In *International Semantic Web Conference, Sardinia*, 2002.

D-FOAF

Distributed Identity Management based on Social Networks

Sebastian Ryszard Kruk
WETI, Gdansk University of Technology, Poland
DERI, NUI Galway, Ireland
sebastian.kruk@deri.org

Adam Gzella
WETI, Gdansk University of Technology, Poland
DERI, NUI Galway, Ireland
adam.gzella@deri.org

Sławomir Grzonkowski
WETI, Gdansk University of Technology, Poland
DERI, NUI Galway, Ireland
slawomir.grzonkowski@deri.org

ABSTRACT

Contemporary Web consists of more than just information, it provides a large number of services, which often require identification of it's users. Since distributed or shared identification systems are not yet widely adopted many users have to maintain a large number of different credentials for different services. Furhermore current authorisation systems require strict centralisation of the authorisation procedure. Although the feature of enabling user's friends or good friends of a friends to access user resources would be benefical for services and business on the Web, it is not usually offered by existing systems. In this article we present D-FOAF, a distributed identity management system that utilizes social networks. We show how information inherent in social networks can be utilised to provide community driven access rights delegation and distributed authorisation.

Keywords

Identity Management, Distributed Computing, Social Networking

1. INTRODUCTION

The proliferation of Internet services introduced many problems like no single identity for Internet users or no scalability in trust and access rights management. Some of those problems have been addressed in many ongoing projects.

The main difference between internet and real world services are authorisation procedures. In the real world each person has a single identity expresses with credentials like an ID card, a passport or a driving licence. This allows real world service providers to easily confirm the authentity of the presented credentials. In the Internet, each user has to deal with a number of identities with different credentials like login-password pairs. Since there is no notion of single identity, service providers are usually inclined to introduce new credentials for each user. As a result the trust to each user is build within each service separately.

Approaches like Microsoft Passport [2], Sxip [7] or Liberty Alliance Project [3] are aiming to provides a solution to the single-sign-on problem. Due to various problems. none of those projects has been widely addopted by service providers so far. So they are useless for the majority of Internet users with the ever growing number of service.

Most of online services are usually based on very simple user profile management implementations that do not address problems stated above. Access rights are based on predefined, fixed lists of groups and neither allow finer granularity nor trust delegation.

The notion of social networking emerged in the Internet with online community portals like Orkut [10] that allow users to control access to the information based on the structure of the social network. Each user can restrict access to some parts of his/her profile information delegating trust within given number of degrees of separation.

Some of problems encountered in Microsoft Passport [2] were overcome by Sxip [7], which enable users to gain more control over their profile information stored on one of home servers. Next version of Sxip 2.0 will provide increased anonymity for users with the Identity 2.0 [1], a protocol for exchange of digital identity information. The general idea is to provide users with more control over what others know about them. Furthermore, it will be possible to adjust security needs to the specific site.

In this short article we introduce the main requirements of the Identity 2.0 protocol (see 2). We present how some of them, including support for mobile computing paradigm, have been implemented in D-FOAF (see 3). Finally we describe an overview of the demo we would like to present at ESWC2006 (see 4).

2. TOWARDS IDENTITY 2.0

Nowadays we have as many identities as services we use. Usually, we can hardly transfer the digital identity from one website to another. Thereofre the trust we gain in one community is useless in another because it does not affect our reputation there. Thus we are forced to repeat the whole process of gathering trust again and again.

Since the contemporary systems does not allow users to decide which information is available for the other portal users, it causes the lack of privacy control over the user profile.

These problems should be resolved by means of an advanced social network system that would model the social interaction close enough to a real world ones. In addition,

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006
.

the solution must be easy to use and enable users to share the credentials among many services. Although the strong security support to protect identities of the users must be provided, all of these features should be transparent from the end-user perspective.

The main objective of Identity 2.0 protocol is to provide users with full control over their virtual identities. It seems to be a suitable solution for the described problem. The presented below the FOAFRealm(see section 3) system aims to meet the requirements of the Identity 2.0 and extend them.

3. DISTRIBUTED FOAFREALM

The FOAFRealm [9] is a library for user identity management based on the FOAF vocabulary [4]. FOAFRealm enables users to control their profile information defined in the open FOAF metadata. To provide enhanced resource rights management FOAFRealm extends generic FOAF description by friendship evaluation based on reification of <`foaf:knows`> RDF statements.

Generic FOAFRealm consist of three general parts:

- FOAF metadata and collaborative filtering ontology management. It wraps the actual RDF storage being used from the upper layers providing simple access to the semantic information. The Dijkstra algorithm for calculating distance and friendship quantisation is implemented in that layer.
- Implementation of the `org.apache.catalina.{Realm, Valve}` interfaces to easily plug-in the FOAFRealm in to Tomcat-based web applications. It provides authentication features including autologin based on Cookies.
- A set of Java classes, Tagfiles and JSP files plus list of guidelines that can be used while developing user interface in own web applications

D-FOAF (Distributed FOAFRealm) [6] project aims to make the FOAFRealm work in fully distributed environment. A new distributed communication layer introduced in D-FOAF provides access to highly scalable HyperCuP Lightweight Implementation [11, 8] of P2P infrastructure to communicate and share the information with other FOAFRealm implementations.

The most important D-FOAF features are:

- Distributed user authentication - realization of Single-sign-on conception.
- Distributed user identity merging - ability to merge distributed user identity on demand.
- Computing distance and trust levels between users in distributed environment.
- Security of distributed computing - creating suitable identity protection.

Next step in FOAFRealm/D-FOAF development is DigiMe, a ubiquitous indentity management compliant with Identity 2.0 assumptions. We made first steps towards DigiMe, by building ubiquitous search and browsing application [5]. It was developed on J2ME platform, and provides simple access to FOAFRealm/D-FOAF identity for mobile devices.

FOAFRealm/D-FOAF system has been successfully deployed with JeromeDL - semantic digital library. In addition to unique distributed identity management FOAFRealm allows JeromeDL to integrate user and author list in semantic query expansion algorithm.

4. PRESENTATION PLAN

During the demo session we will present how distributed identity management based on social networking works in existing systems and how it can be easily deployed in new services. The demonstration will consist of:

1. Generic FOAFRealm in JeromeDL: registering new user, logging into library, adding friends.
2. D-FOAF's distributed authentication: logging into different JeromeDL instances using existing set of credentials.
3. D-FOAF's user identity merging: gathering distributed user identity in JeromeDL.
4. D-FOAF's distance and quantisation level computing: accessing the protected resource using the friedship informations saved in different JeromeDL instance.
5. DigiMe mobile application: managing identity information, friends list and bookmarks using mobile device.

4.1 Acknowledgments

This work was supported by Science Foundation Ireland Grant No. SFI/02/CE1/I131 and by the Knowledge Web project (FP6 - 507482) and partially by KBN, Poland under grant No. 4T11C00525. The authors would like to acknowledge Stefan Decker, John Breslin, Tomasz Woroniecki, Choi Hee Chul, the DERI Semantic Web Cluster and the Corrib.org working group for fruitful discussions.

5. REFERENCES

[1] Identity 2.0: http://www.identity20.com/.
[2] Microsoft Passport: http://www.passport.net.
[3] L. Alliance and WS-Federation. A Comparative Overview. White Paper. Technical report, 2003.
[4] L. Dodds. An Introduction to FOAF. http://www.xml.com/pub/a/2004/02/04/foaf.html, 2004.
[5] S. Grzonkowski, A. Gzella, M. Cygan, and S. R. Kruk. Digime - ubiquitous search and browsing for digital libraries. In *Submitted to MoSo 2006 Workshop*, 2006.
[6] S. Grzonkowski, A. Gzella, H. Krawczyk, S. R. Kruk, F. J. M.-R. Moyano, and T. Woroniecki. D-foaf - security aspects in distributed user management system. In *TEHOSS 2005*, September 2005.
[7] D. Hardt. Personal Digital Identity Management. In *FOAF Workshop proceedings*, 2004.
[8] HyperCuP Lightweight Implementation project: http://www.hypercup.org/.
[9] S. R. Kruk. FOAF-Realm - control your friends' access to the resource. In *FOAF Workshop proceedings*, http://www.w3.org/2001/sw/Europe/events/foaf-galway/papers/fp/foaf_realm/, 2004.
[10] Orkut. http://www.orkut.com/.
[11] M. Schlosser, M. Sintek, S. Decker, and W. Nejdl. HyperCuP–Hypercubes, Ontologies and Efficient Search on P2P Networks. In *Third International Workshop on Agents and Peer-to-Peer Computing*, 2004.

MarcOnt Portal

Ubiquitous Collaborative Ontology Life-cycle Management

Sebastian Ryszard Kruk
DERI, NUI Galway
IDA Business Park
Lower Dangan, Galway, Ireland
sebastian.kruk@deri.org

Paweł Szczecki
WETI, Gdansk University of Technology
Narutowicza 11/12
Gdansk, Poland
pawel.szczecki@deri.org

Michał Wozniak
WETI, Gdansk University of Technology
Narutowicza 11/12
Gdansk, Poland
michal.wozniak@deri.org

Piotr Piotrowski
WETI, Gdansk University of Technology, Poland
DERI, NUI Galway, Ireland
piotr.piotrowski@deri.org

ABSTRACT

The Semantic Web effort, which partially originated from the digital library community (Dublin Core), is providing technology such as ontologies that can be potentially applied to the problem of managing resources. The goal of the MarcOnt Initiative is to create a new bibliographic description standard in the form of an ontology and related tools utilizing semantic technologies.

Building an ontology should be an effort of the community of domain experts - librarians in the case of MarcOnt ontology. Therefore one of the goals of MarcOnt Initiatives was to build a community portal for building a social ontology by means of negotiations and versioning. In this paper we present MarcOnt Portal a new approach to a collaborative ontology management. The MarcOnt Portal is a framework for storage, versioning and development of the MarcOnt ontology together with mapping rules (between commonly used bibliographic formats and MarcOnt). Although MarcOnt portal has been built especially for the MarcOnt ontology it is open to be used in any OWL-based ontology development endeavour.

Keywords

Ontology Management, Collaborative Environment, Negotiations

1. INTRODUCTION

Creating an ontology for any specific domain of discourse is time consuming and requires expertise in several fields which might surpass the capabilities of one person alone. If this is the case there is the need for collaboration effort of many people, so called *domain experts*. Domain experts participating in such a task might be geographically distributed. Thanks to the Internet this work can be done from any computer in the world. So the best way to help many people work on such a task is to create a kind of a portal that will help them communicate, negotiate and cooperate. Additionally since some of domain experts are traveling very often, they would facilitate from ubiquitous access to the ontology management portal.

There are many tools for editing ontology [1]. Some of them, like Protege [4] or Ontolingua [2], support collaborative work. Recently emerging initiatives like Peoples Portal [9] or DOME [3] aim to deliver environments where domain experts may engineer ontologies in a community fashion.

Existing solutions allow for a group of people to simultaneously work on one ontology, but unfortunately they usually stored no versioning information or users are allowed to simply change what they want. This is sufficient for a small communities portals where quality and the proper lifecycle of ontology management is not a key aspect.

In this short article we will briefly present the idea of the collaborative ontology management portal that has been realized in MarcOnt Portal project (see 2). We will define main components of MarcOnt Portal (see 3) and describe an overview of the demo we would like to present at ESWC2006 (see 4).

2. COLLABORATIVE ONTOLOGY LIFE-CYCLE MANAGEMENT

MarcOnt Initiative [7] defined requirements for bibliographic ontology evolution. Since bibliographic descriptions already exist in many popular formats like MARC21, BibTeX or DublinCore, therefore the process of developing an appropriate ontology should bring together different domain experts. The environment should allow them to define rules for translating to and from newly defined concepts. Domain experts should develop an ontology by interacting with each other's suggestions by means of negotiations. Voting on changes

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006
.

made by others domain experts ensures that the best solution according to the whole community will be introduced to the next version of the ontology. The full lifecycle of ontology development requires also versioning of the ontology itself and domain experts' suggestions as well. There are several solutions that are ready to use like changes tracking in RDF repository (Sesame[1]) or some dedicated RDF version solutions like SemVersion [8].

As it has been already mentioned sophisticated task of developing an ontology requires work of many people. Therefore, the ontology creation lifecycle had to be split into several distinct parts to preserve the flexibility of collaborative development and to simplify the decision process. The lifecycle goes as follows [5]:

1. The initial revision of the ontology is presented in the portal.
2. Domain experts can submit their suggestions which are added into the tree of suggestions.
3. Each user can view other users' suggestions, edit them and add as their own suggestion. Many users can work on the same suggestion at once within one session. Suggestions can be voted for and against.
4. When the suggestion is considered to be mature enough, a final vote is held and the suggestion is included into the main ontology.
5. At the same time, a semi-automated process of conflict detection is run by the portal service. If any conflicts are found, suggestions are held back until they will be redesigned to avoid conflict.
6. When a suggestion meets all the required criteria (conflictless and votes) it is merged with the main ontology and a new revision of the ontology is created.

3. MARCONT PORTAL IMPLEMENTATION

Following requirements defined in the MarcOnt Initiative [5] on the collaborative environment for ontology management (see 2) we have built the prototype of the MarcOnt Portal as a web application with some mobility components:

The Repository Component is a starting component for further work with the portal. It gives the graphical access to the whole repository including suggestions and their versions, main ontology and the mapping rules. When an object is selected from the repository this component loads either ontology editor or rules editor depending on the selected object. Additional functionality provided by this component is semantic diff view between any two suggestions and any two versions, creation of an empty ontology and uploading existing OWL file as a new suggestion. The repository component hold the information not only about repository but also about the users.

The Editor Component consists of three separate tabs. *Classes tab* brings ways to manage classes (adding, editing, moving, removing, defining sub/superclass relations, setting equivalents and disjoints, comments); *Properties tab* enables properties management (adding, editing, moving, removing, defining domain and range, defining sub/superproperties relations, setting equivalents and disjoints, comments); *Namespaces tab* allows adding namespaces used in the classes and properties tabs

The mapping rules editor component . Current implementation allows to load two ontologies (in OWL format) and a set of rules (in XML or RDF format). User interface, both web and mobile [6], allows to select concepts from both ontologies to be used in rules development. Domain expert can define premises and consequents based on ontology concepts, variables and regular expressions. The interface has been already evaluated against MarcOnt and MARC21 ontologies with mapping rules from one to another.

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

4. PRESENTATION PLAN

We would like to present the MarcOnt Portal according to following plan:

1. Loading new ontology;
2. Commit the ontology as a new suggestion;
3. One domain expert is suggesting some changes;
4. The changes are visualized;
5. Commit the changes as a new version of the suggestion;
6. Another domain expert is suggesting some other changes;
7. View the repository and the semantic diff of the two suggestions;
8. Loading two ontologies into the rule editor and generating a suggestion of a new translation rule;
9. Performing rules management with a mobile device;

4.1 Acknowledgments

This work was supported by Science Foundation Ireland Grant No. SFI/02/CE1/I131 and by the Knowledge Web project (FP6 - 507482) and partially by KBN, Poland under grant No. 4T11C00525. The authors would like to acknowledge Stefan Decker, the DERI Semantic Web Cluster and the Corrib.org working group for fruitfull discussions.

5. REFERENCES

[1] Duineveld, A. J., Stoter, R., Weiden, M. R., Kenepa, B., and Benjamins, V. R. Wondertools: A comparative study of ontological engineering tools. *International Journal of Human-Computer Studies* (June 2000), 1111–1133.

[2] Farquhar, A., Fikes, R., and Rice, J. The ontolingua server: A tool for collaborative ontology construction, 1996.

[3] Henke, J. The table metaphor: A representation of a class and its instances. In *Proceedings to Workshop on User Aspects of the Semantic Web* (Heraklion, Greece, may 2005).

[4] Knublauch, H., Fergerson, R. W., Noy, N. F., and Musen, M. A. The protg owl plugin: An open development environment for semantic web applications. In *Third International Semantic Web Conference - ISWC 2004* (Hiroshima, Japan, 2004).

[5] Kruk, S. R., Synak, M., and Zimmermann, K. Marcont initiative. bibliographic description and related tools utilising semantic web technologies. Tech. rep., DERI, NUI Galway.

[6] Piotrowski, P., and Kruk, S. R. Ubiquitous ontology lifecycle management. In *Submitted to MoSo 2006 Workshop* (2006).

[7] Sebastian Ryszard Kruk, Marcin Synak, K. Z. Marcont - integration ontology for bibliographic description formats. In *Proceedings to DublinCore Conference 2005* (2005).

[8] Völkel, M., Enguix, C. F., Kruk, S. R., Zhdanova, A. V., Stevens, R., and Sure, Y. Semversion - versioning rdf and ontologies. Knowledge Web Deliverable 2.3.3.v1, University of Karlsruhe, JUN 2005.

[9] Zhdanova, A. V. The people's portal: Ontology management on community portals. In *Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web (FOAF'2004)* (Galway, Ireland, September 2004), pp. 66–74.

A Demonstration of Adaptive Work-Centered User Interface Technology

Amy Aragones
GE Global Research
1 Research Circle
Niskayuna, NY
518-387-4009
aaragones@crd.ge.com

Jeanette Bruno
GE Global Research
1 Research Circle
Niskayuna, NY
518-387-6009
bruno@crd.ge.com

Andrew Crapo
GE Global Research
1 Research Circle
Niskayuna, NY
518-387-5729
crapo@crd.ge.com

Marc Garbiras
GE Global Research
1 Research Circle
Niskayuna, NY
518-387-6654
garbiras@crd.ge.com

ABSTRACT

A key challenge in designing for the Semantic Web is to address open-ended decision-making scenarios. This demonstration will show the benefits of Adaptive Work-Centered User Interface Technology (ACUITy)in this regard in the context of a Professor/Student Course Management application.

Keywords

Semantic Web, Semantic Technology, Intelligent User Interface, Adaptive User Interface, Web Portal, Work-Centered Support System

1. INTRODUCTION

One of the challenges of designing applications for the Semantic Web is to create mixed interaction structures in which users and agents can collaboratively solve problems that may use a multitude of alternative, often ad-hoc decision-making strategies. The appropriateness of alternative strategies might be influenced by subtle differences in user goals, preferences and/or other context that is difficult to anticipate in the upfront design of a new web application. A second related challenge is to create an environment where new Semantic Web applications can easily be developed, particularly designing for these open-ended decision-making scenarios.

Our demonstration introduces the Adaptive Work-Centered User Interface Technology (ACUITy) decision support environment, which begins to answer these challenges. In ACUITy we use semantic models to characterize the users' work domain in terms of "work-centered"[1] activities as well as the interface and interactions between the decision support system and the user. We use these semantic models to provide adaptive interaction, both user directed and automated, in the characterization and presentation mechanisms of a web-based user interface.

The Professor/Student Course Management (PSCM) application built using ACUITy will illustrate the use of semantic technology to implement work-centered decision support and the benefits of ACUITy to both application users and developers. We will use this demonstration to highlight the broader potential for ACUITy in areas that are of interest to the Semantic Web community. The PSCM application will also accompany our release of ACUITy on open source in 2006.

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 11th - 14th June, 2006

2. PROFESSOR/STUDENT COURSE MANAGEMENT APPLICATION

2.1 Professors' Course Management

Our demonstration focuses on two aspects of an instructor's course management work. First, we will demonstrate enhanced capabilities for monitoring and understanding the logistical aspects of a course, including the size of the course, sections, schedules, support staff and the publishing of the syllabus, including evaluation and grading formula for the course.

Second, we target the analysis and production of student grades at the end of the semester. This includes supporting the assessment of the class numerical grades (as computed by the formula published at the beginning of the semester) and establishing the numeric breakpoints for translating the students' grade averages into an A, B, C, D, or F grade.

2.2 Students' Course Management

We will also demonstrate support for a portion of the work students perform in creating a course schedule. Before a new semester begins the student must determine which courses to take and register for them. This usually involves understanding the student's course requirements and academic status relative to the University's degree requirements. It also requires understanding course offerings with respect to fulfilling the student's degree requirements and scheduling constraints.

The following sections elaborate on the capabilities we will demonstrate.

3. WHAT ACUITy MEANS TO USERS

3.1 Users Finish the Design

In an ACUITy application the users themselves finish the design of a user interface by deciding what information they require to solve a particular problem – defining the vantage they need on the problem domain – and changing the characteristics of the information display in order to interact with the data more effectively. ACUITy captures in a centralized way the experience of users in open-ended problem-solving domains as they gather information from many disjoint sources not precisely identified at design time.

The user can reconfigure the display by adding and removing displayed information. The approach can be extended to permit ad-hoc additions of information sources. Customization of information display includes, but is not limited to, the hiding, ordering, and sorting of data table columns, the selection of graph series types, e.g., line versus bar, color, and labels, and the type of enumerated selection lists, e.g., dropdown list versus checkbox versus tabs. The user can duplicate and then modify visualizations as desired. Information in disjoint tables and graphs can also be brought into relation by creating shared highlight regions, similar to data brushing in statistical graphics.

3.2 Learning Defaults and Patterns

Learning from accumulated instance data is an implicit benefit of the semantic modeling approach taken by ACUITy. As users adapt the content and visual characteristics of the information that they view in particular problem-solving settings, these changes are stored in the ontology with their context. This past history creates the opportunity for a reasoner to infer what information content is most appropriate based on new information that was unavailable during the initial design of the user interface. This special purpose reasoner then uses the instance data to learn both default content and appearance for new sessions with similar contexts. We have implemented several different algorithms to learn new default displays from instance data. We will demonstrate the effects of these algorithms and how we have implemented them using OWL.

While not yet implemented in ACUITy, recognizing beneficial patterns of usage across groups of users can lead to new classes of display objects explicitly available to developers and users, whereas learned defaults are only implicitly available. It is not inconceivable that abstraction of useful patterns might even extend across application domains.

4. WHAT ACUITy OFFERS DEVELOPERS

The ACUITy Problem-Vantage-Frame (APVF) ontology, described below, provides developers of new applications a starting point from which they can create information-rich displays by relatively simple model extensions. With respect to the user-interface, the developer is also "finishing the design." For example, a new data table can be added to a display through a few simple steps. The behavior and attributes necessary for the table to be constructed and displayed, as well as those that allow the user to customize the table display according to their particular preferences, are inherited.

Domain-specific work models also utilize upper-level ontologies in ACUITy. These ontologies define concepts of time, physical versus abstract, problems, scripts, processes, and remote data sources. Scripting capability includes support of custom Java code that can implement data access, data transformation, or side effects. This facilitates integration of ACUITy applications with existing information repositories and computational models.

5. IMPLEMENTATION APPROACH

ACUITy has three major components:

1. *The ACUITy Problem-Vantage-Frame (APVF) ontology*, which, situated within a hierarchy of upper level and domain-specific ontologies, represents concepts and properties that describe users, the problems they are trying to solve, the information they (and other users) have used to solve those types of problems and the display properties of that information. The APVF ontology is represented in OWL.
2. *The ACUITy Controller*, a Java class (with supporting classes) that provides an API to the APVF ontology. It provides special-purpose reasoning over this knowledge base to determine the set of information relevant to the problem at hand or the context of work performed. The ACUITy controller queries the ontology to understand where to find data, how to obtain it, and how to bundle it. The controller also accepts inputs from the client UI engine and updates the ontology accordingly.
3. *The User Interface (UI) Engine*, which accepts the ontological information obtained from the ACUITy Controller and creates the application's user interface. It interacts with the controller to request information from the ontology in response to the user's actions. At this point, we have implemented a web client renderer to produce a well-formed HTML document from the UI engine.

6. FUTURE DIRECTIONS

To date, ACUITy has been used to prototype several web applications of interest to General Electric, Lockheed Martin and the US Air Force. Our preliminary experience is that it is a powerful and very flexible environment for developing decision support systems. Additional user testing and developer feedback is needed to validate this initial assessment.

We plan to create - and hope to also demonstrate at ESWC - an ACUITy Ontology Editor that will simplify the process of creating and editing web applications for both developers and end users. New sources of data will become relevant to users during the course of their work, potentially via web-services, ad hoc web searching and other means of accessing information. We would like users to be able to map to these sources of information in real time.

We hope that by making this work available to the public we can contribute to the wide-spread use of semantic technology. We also believe that wider research efforts will be able to leverage data-rich models created by ACUITy that are populated with semantically tagged information. These will provide an important resource for further research in semantically-enabled UI and decision support system design, adaptation and learning.

7. ACKNOWLEDGMENTS

This work was partially funded by the Air Force Research Laboratory, Wright Patterson AFB, under contract F33615-03-2-6300. We would like to acknowledge the contributions of AFRL and Lockheed Martin STS, who provided support to and feedback on this research.

8. REFERENCES

[1] Eggleston, R.G., Young, M.J., and Whitaker, R.D. Work-Centered Support System Technology: A New Interface Client Technology for the Battlespace Infosphere. *NAECON 2000*, 499-50.

iFanzy: Personalised Filtering Using Semantically Enriched TV-Anytime Content

Paul Akkermans
Eindhoven University of Technology
P.O.Box 513
The Netherlands
p.j.j.akkermans@tue.nl

Lora Aroyo
Eindhoven University of Technology
P.O.Box 513
The Netherlands
l.m.aroyo@tue.nl

Pieter Bellekens
Eindhoven University of Technology
P.O.Box 513
The Netherlands
p.a.e.bellekens@tue.nl

ABSTRACT

The ongoing media digitalization is creating opportunities to bring new interactivity to the traditional TV concepts. The XML-based TV-Anytime standard for TV content description is tightly coupled to the MPEG-7 ontology. We translated the TV-Anytime ontology to OWL by making use of an existing OWL version of MPEG-7. We defined mappings to existing ontologies for time, geography and linguistic concepts. The demonstration of the iFanzy personalized electronic program guide shows our ontology-based approach to personalized access to TV content considering the user context and providing semantically-meaningful recommendations to viewers. The approach involves proper modeling of the domain and of the additional knowledge that is included in the system, as well as data transformations that match the ontological knowledge.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Interactive TV

Keywords

Semantic annotation, multimedia, electronic program guides, personalization

1. INTRODUCTION

While the Web steadily but continuously keeps its advance towards a full-blown user-adaptive content collection, other similar content providers like television broadcasters are lagging behind. Various studies show the need for personalization in dealing with the massive TV content [6]. Content providers search for ways to put all their content at a digital visitor's disposal. However, this creates new demands regarding personalization [2], handling diversity of users and/or groups of users, interaction and content explosion issues [1] on an even more diverse scale. The process of integrating content collections from different heterogeneous sources and presenting them to the users in a personalized and context-aware manner demands a good understanding of both the content we are dealing with and the users using it [3]. In this paper we concentrate on the use of ontology-based knowledge in enriching the personalized interaction with content collections. We introduce the iFanzy personalized electronic programming guide for ambient home media environment. It is a collection of filters for retrieving and presenting incoming TV content according to user preferences, characteristics and contexts for TV viewing. It is designed and implemented in compliance with the TV-Anytime-OWL based architecture of the Blu-ray Interactive System. The demonstrator presented in this paper is a collaboration between Eindhoven University of Technology, Stoneroos Interactive Television and Philips NL in the context of ITEA funded Passepartout project.

2. BLU-RAY INTERACTIVE SYSTEM

The software called Blu-ray Interactive System (Blu-IS) illustrates an ambient home media environment to enable ontology-based personalized access and interaction with digital TV content. Blu-IS is a connecting point for home devices, such as shared (large) screens, personal (small) handhelds, hand-gesture recognition and biosensor-based interfaces. The Blu-IS system is responsible for the personalization of the user-content interaction satisfying the diverse requirements of different users, and intelligent information filtering in order to prevent an information overflow as the abundance of available TV content will be very large. Fundamental in our approach is the use of ontology-based modelling of the media content and user information in order to incorporate ontological background knowledge in the user's access to the content collection 1. In this way, we achieve optimal expressivity and semantic relations of the TV content. We have translated the TV-Anytime classification into an OWL ontology and realized mappings to time [4] and GeoNET [5] ontologies in order to achieve more dynamic handling of the otherwise static TV content descriptions, and provide users with flexible composition of relevant content packages. The use of lexical thesauri, such as WordNet, allows us also to refine user queries with synonyms and other word forms. When the user posts a query, the Blu-IS uses

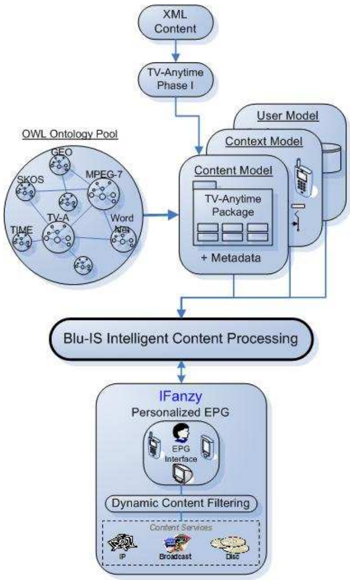


Figure 1: Blu-IS

Figure 2: iFanzy Screenshot

the ontological knowledge, in terms of background, context and user knowledge, and constructs a corresponding refined query to the content repository. Thus, it adapts the result to the needs of the user or group of users and ranks them according to their relevance. For example, the user can ask for all the content available in a particular time frame, specific location or on a preferred topic. After filtering the relevant content it is presented to the user in a packaged form, including not only the TV programs, but also related content from the Web, such as soundtracks, posters, pictures, etc.

3. IFANZY PROGRAM GUIDE

The iFanzy electronic program guide 2 uses the Blu-IS semantic-based information management for the realization of TV content filters considering the user and user's context.

It allows XML-based content to be mapped to the TV-Anytime metadata schema and further organized in TV-Anytime content packages. iFanzy is developed as a Java application currently using real BBC program stream provided by the BBC web site. It consists of a set of filters, which can be extended depending on the demand for filtering criteria. Currently, we have developed the following filters 3:

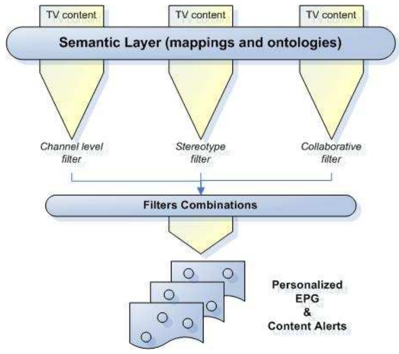


Figure 3: iFanzy Filters Overview

- Channel level filter: The user can add/delete a channel to the preferred set of channels. Channels not contained in the set of preferred channels are not shown in the iFanzy.
- Stereotype filter: Every user is matched to a set of stereotype users. Every stereotype group of users has their own preferences and viewing behavior.
- Collaborative filter: Based on the viewing history of the user, the user is matched to a set of other users with the same interests and preferences.

4. CONCLUSIONS

The goal of the iFanzy demonstrator is to how semantic annotation of TV content and modeling of domain, user and context can provide an efficient alternative to the existing theme channels - dynamic composition of TV content packages based on data semantics and user profile and context.

5. ACKNOWLEDGMENTS

This research is supported by the European ITEA Passepartout project. Special thanks to Rop Pulles, Peter Hulsen from Philips NL, and Geert-Jan Houben, Tim Dekker and Erik Loef from TU/e.

6. ADDITIONAL AUTHORS

7. REFERENCES

[1] Web engineering, software engineering and web application development. In S. Murugesan and Y. Deshpande, editors, *Web Engineering*, volume 2016 of *Lecture Notes in Computer Science*. Springer, 2001.

[2] R. Denaux, V. Dimitrova, and L. Aroyo. Integrating open user modeling and learning content management for the semantic web. In *User Modeling Conference*, 2005.

[3] N. Henze. Personalization functionality for the semantic web: Identification and description of techniques. Technical report, REWERSE EU NoE, 2004.

[4] J. R. Hobbs and F. Pan. An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(1):66–85, 2004.

[5] http://earth-info.nga.mil/gns/html/. *GEOnet Names Server (GNS)*.

[6] A. K. Liliana Ardissono and M. Maybury, editors. 2004.

OnTour: Tourism Information Retrieval based on YARS

Katharina Siorpaes
Digital Enterprise Research Institute
Technikerstraße 21a
Innsbruck, Austria
+43 512 507-6488
katharina.siorpaes@deri.org

Daniel Bachlechner
Digital Enterprise Research Institute
Technikerstraße 21a
Innsbruck, Austria
+43 512 507-6488
daniel.bachlechner@deri.org

ABSTRACT
Semantic Web enabled information retrieval can help to overcome problems caused by information flooding in the World Wide Web. Various industry sectors are faced with the challenge to set up and maintain websites that provide information required by the consumer. Within the scope of the OnTour project, a system based on a fast and flexible Semantic Web backbone has been developed focusing on e-tourism. The major benefits of the OnTour approach, such as its simplicity, modularity, and extensibility, can already be highlighted with the help of the system's current prototype.

Categories and Subject Descriptors
H.3.3 [**Information systems**]: Information Search and Retrieval – *Clustering, Information filtering, Query formulation, Relevance feedback, Retrieval models, Search process, Selection process*

General Terms
Experimentation

Keywords
Semantic Web, RDF, YARS, Information Retrieval, Tourism

1. INTRODUCTION
Ontologies and ontology-based information retrieval have the potential to significantly improve the process of searching information on the World Wide Web. Concept search and browsing can ease the burden of searching the web using keyword based techniques. This is especially important in information-based industries, such as e-Tourism. The predominant goal of the OnTour project is to show that the application of Semantic Web technologies in a real-life scenario can have substantial effects on current flaws such as information flooding in Web-based tourism environments. The OnTour system is a starting point for a field study in cooperation with the tourism industry in Austria to investigate the application of Semantic Web technologies in a real-life scenario. The affinity to new information technologies and the increasing importance of the internet make e-tourism a perfect candidate for this study. There are many challenges but only a small selection that have major impact on the tourism industry and are therefore relevant for the OnTour project is cited here as examples. In this demo proposal, we describe a first prototype based on the RDF store YARS. The system's architecture is accentuated in section 2. In section 3, we explain how the demonstration itself will be conducted. In section 4, we provide an outlook to future work. Finally, we present our conclusions in section 5.

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 11th – 14th June, 2006.

1.1 Major Challenges
The tourism industry has an above-average growth rate as well as a great economical importance throughout Europe. It is notably information dependent and needs new ways to deal with its vitality and scale. At present primary tourism products entail high information search costs for the clients as well as the suppliers. According to [4] such informational market imperfections lead to the establishment of long and costly information and value chains. In the highly competitive European tourism market it is vital for primary suppliers to provide a fair amount of information online. Due to the SME structure of the European hotel and restaurant sector, which includes roughly 8.5% of all European enterprises, it isn't affordable for most of the suppliers to set up, maintain, and advertise their own websites. A study in the tourism market described in [2] shows that only 60% of accommodations maintain their own websites, while 93% are members in tourism portals. [3] points out that currently the only alternative for primary suppliers to participate in e-tourism are structured information repositories which are built and maintained at high costs mostly from large providers of all inclusive packages. But besides the high costs, further problems such as long winded data maintenance, low adaptation rate and general inflexibility as well as suboptimal target groups argue against it.

1.2 Proposed Solution
OnTour[1] is a lightweight system based on the RDF data store YARS [1]. The approach ultimately makes it possible to negotiate a large part of the major restraints known in connection with Web-based tourism and in particular the challenges mentioned above. YARS relies on optimized index structures and thus enables fast storage and retrieval of large amounts of data while retaining the small footprint and lightweight architecture approach of OnTour. Describing data in RDF helps to overcome a list of problems known in Web-based information systems with respect to quality and processability.

[1] http://ontour.deri.org

2. SYSTEM ARCHITECTURE

OnTour enables querying a set of accommodation facilities of different types and categories by specifying a list of constraints. The primary user interface is Web-based. In this section, we describe the user interface, the data store, and the internal commutation of the system.

2.1 User Interface

The Web interface provides the user with a form that allows them to restrict the set of results by filtering them according to location and type constraints. For the purpose of debugging the query sent as well as the results received from the data store are displayed in text areas. The result values can be specified before submitting a query by checking the particular boxes. The result can also be shown in XHTML at the bottom of the page. In this case all result values are displayed by default. XSLT is used to transform the result returned by the data store according to the guidelines defined in the XSL files. With the help of XSL and CSS files the actual output can be defined and adjusted to respective needs easily with-out having to change or even understand the rest of the code. Ensuring the ease of OnTour's use has always been an important factor during the development process of the user interface. The context-aware form adapts automatically without reloading the page to changes in the task being performed. Intuitive handling as well as fault tolerance have always been major issues and will become even more important as the prototype is evolving.

2.2 Data Store

YARS is a data store for RDF which allows querying based on a declarative query language. The YARS servlet uses RDF/N3 for encoding facts as well as queries and provides an HTTP interface for performing query, add, and delete operations. By default YARS delivers results in form of RDF/N3 but the HTTP Accept Header can be used to make the YARS servlet return in XML format. YARS is a system that combines methods from information retrieval and databases to allow better query answering performance. In contrast to other implementations such as Jena2 and Sesame, YARS provides optimized index structures. The data set in use for the current version of the prototype including more than 4600 Tyrolean hotels was kindly made available by the Austrian Federal Economic Chamber. The data structure is very simple and comprises only two concepts, accommodation facilities and related occupational groups.

2.3 Communication

The N3QL queries are assembled dynamically on the client’s side before they are sent to YARS using the XMLHTTP API. The communication between the Web interface and YARS happens asynchronously to ensure maximum usability. The HTML presentation is directly connected to XML data for interim updates without reloading the page. The results, delivered in XML, are further processed and displayed on the client's side.

3. DEMONSTRATION

The demonstration of the OnTour system consists of two parts. First the architecture is explained and visualized, second the functionality of the prototype is demonstrated. A variety of queries are executed and their results discussed during the demonstration. Furthermore the advantages of the approach at hand are brought out clearly from both a tourist's and a tourism operator's point of view. Within the scope of the demonstration it is also shown how the major challenges described in the introduction can be overcome with a fully-fledged tourism information system based on the notions of OnTour.

4. FUTURE WORK

To show the full potential of the RDF-based OnTour system, associate information has to be combined with the existent data sets. Geographical information is a possibility as well as information about classification criteria, leisure infrastructure or events. In the medium run information depth, scalability and performance have to be analyzed and optimized. This happens hand in hand with substantial extensions of the knowledge base in terms of the quantity and quality as well as the necessary adaptations concerning the user interface and its usability. The modular design and the simplicity of the prototype guarantee easy exchange and extension of components in the future. The lightweight architecture also allows the system's adaptation to the requirements of newly emerging standards and devices.

5. CONCLUSION

E-tourism is a perfect candidate for Semantic Web because it is information-based and depends on the WWW, both as a means of marketing and transaction channel. Ontology-based information retrieval makes it possible to handle the known challenges in connection with Web-based information systems in a more efficient way. Even though OnTour is a prototype carrying a relatively small stock of data based on a small ontology, it is able to demonstrate its use very well and more importantly can give an interesting outlook on future work.

6. REFERENCES

[1] Andreas Harth, and Stefan Decker. Yet Another RDF Store: Perfect Index Structures for Storing Semantic Web Data With Contexts. Research Paper, 2005.

[2] Martin Hepp, Katharina Siorpaes, and Daniel Bachlechner. Towards the Semantic Web in E-Tourism: Can Annotation Do the Trick?. In *Proceedings of the 14th European Conference on Information System (ECIS 2006)*, Gothenburg, Sweden, June 12-14, 2006.

[3] Alexander Mädche, and Steffen Staab. Applying Semantic Web Technologies for Tourism Information Systems. Research Paper, 2001.

[4] Hannes Werthner, and Francesco Ricci. E-Commerce and Tourism. In *Communications of the ACM*, 47(12):101-105, New York, December 2004. ACM Press.

Ontoprocess – a prototype for semantic business process verification using SWRL rules

Hans-Jörg Happel
FZI Research Center for Information Technologies
Haid-und-Neu Str. 10-14
76131 Karlsruhe
Germany
happel@fzi.de

Ljiljana Stojanovic
FZI Research Center for Information Technologies
Haid-und-Neu Str. 10-14
76131 Karlsruhe
Germany
stojanovic@fzi.de

ABSTRACT

In this paper, we describe Ontoprocess, a prototype implementation for semantic business process management (sBPM)[1] consisting of a simple rule editor and a process modelling workbench. Our focus is to provide means for automatically checking the compliance of business processes with business rules. Therefore we combine semantically described business processes with SWRL rules by a set of shared ontologies, capturing knowledge about a business domain. These formal specifications enable us to automatically verify if a process description satisfies the consistency constraints defined by business rules.

1. MOTIVATION

In today's business world, the management and flexibility of services are key success factors for an IT-enabled enterprise. Companies have to assure that their business processes comply with new regulations like the *Sarbanes-Oxley Act* or *Basel II.* Additionally, the possibility for a flexible reconfiguration of services is required for quickly reacting to market changes and customer demands.

These two fundamental requirements – compliance with regulations and flexibly changeable processes – are a big challenge for business process management. In the case of new regulations, all processes have to be revisited in order to assure their compliance. In the case of changing a process, it has to be verified against all regulations. A powerful business process management environment should assist those activities by providing means for automatically verifying the consistency of business processes and guide the process engineer to implement the required changes.

2. SOLUTION APPROACH

In order to automate the verification of business processes, regulations and business processes must exist in a formal, machine-understandable representation. In the following we present a combination of semantic web technologies with process modelling and business rules to tackle this issue.

Our concept involves a two level architecture of process modelling (see Figure 1). The upper layer includes domain information, capturing central business concepts of an organization in ontologies, and business rules in a formal rule representation. The second layer consists of semantic-based process models, describing the organizations' business processes.

Both layers can be maintained by appropriate experts. The process activities are annotated with the domain concepts, thus providing a propagation link in case of changes.

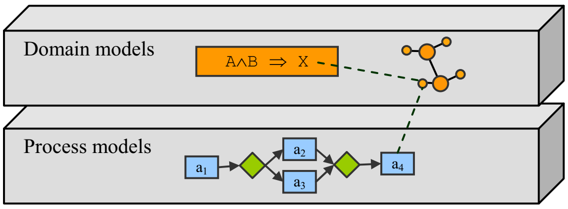


Figure 1: Layers of semantic web-enabled business process modelling

In order to employ formal methods for checking the consistency of business processes, both, the process models describing the actual system behaviour and the correctness requirements need to be specified in a formal language. Thus, we use OWL ontologies for process description and DL-safe rules (a decidable subset of SWRL rules [1]) to express correctness requirements. The process description ontologies are based on an extension of OWL-S[2]. Since the KAON2 inference engine[3] is capable of processing both formalisms, we use it to check if process models satisfy the consistency constraints defined by the rules.

3. EXAMPLE

Our motivating example comes from the area of veterinary regulation. We combine fictitious business processes of a meat processing company with data inspired by an EU regulation that specifies "rules for the organisation of official controls on products of animal origin intended for human consumption"[4].

Therefore we modelled some business processes (such as a procurement process for chicken) and created domain-specific ontologies (e.g. about animals or veterinary regulation) to annotate them. We show how our framework guides the maintenance of business processes in case of adding new or changing existing rules.

[1] http://km.aifb.uni-karlsruhe.de/ws/sbpm2006, http://www.sbpm.org/

[2] http://www.daml.org/services/owl-s/

[3] http://kaon2.semanticweb.org/

[4] http://europa.eu.int/eur-lex/lex/LexUriServ/LexUriServ.do?uri=OJ:L:2004:139:0206:0319:EN:PDF

Our scenario is to demonstrate the change of a business rule and its subsequent propagation and implementation into the process layer. This change shall pertain to an existing rule (see Figure 2). It demands that every process that is annotated to operate within a *Procurement* Context dealing with *Chicken* has also to include a *Visual-check*. In our scenario, we assume a regulatory change that demands a higher-level *Ante-mortem inspection* instead.

```
Process(p) ∧ ProcurementContext(x) ∧
Chicken(c) ∧ hasContext(p,x) ∧
hasSubject(p,c) ∧ ¬R(p) ⇒ Error(p)

coversRegulation(p,v) ∧
Visual-check(v) ⇒ R(p)
```

Figure 2: Example rule

When verifying the business processes in the process modelling workbench, a process *procureChicken* is highlighted. It has become invalid, because the annotated *Visual-check* is no more sufficient in order to comply with the business rules. Changing this to an *Ante-mortem inspection* makes the process valid again. Similarly, consistency checking can highlight invalid activities when adding a new business rule to the rule set.

4. IMPLEMENTATION

A meta-process ontology on top of the domain model and the process model layer provides some basic concepts that are specialized in each layer (see Figure 3). It contains entities used for ontology-based process modelling such as *Activity, Regulation* or *Context* and a set of relations among them, namely *hasContext*, *hasSubject* and *coversRegulation*.

The domain model layer includes domain ontologies, refining concepts from the meta-process ontology in order to have domain-specific data needed for process description. In our scenario this is a *Veterinary* ontology, defining concepts of veterinary regulation, an *Animal* ontology and a *Business Context* ontology containing categories for business processes such as *procurement*.

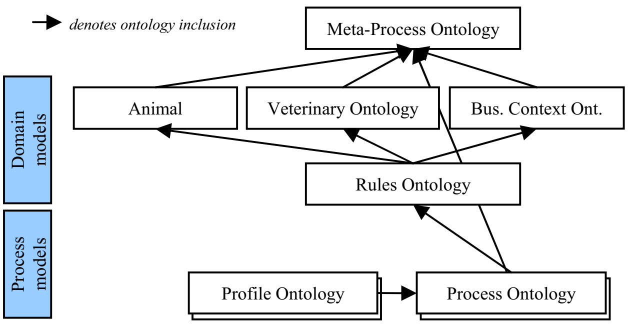


Figure 3: Ontologies in the scenario

Furthermore, a set of rules is part of the domain model layer. It contains business rules (see Figure 2 for an example), relating concepts from the domain-specific ontologies to process activities by the relations defined in the meta-process ontology. We implemented a GUI (“RulEd”) appropriate for end-users to model a simple kind of IF/THEN rules. The rules are internally created using the KAON2 API and saved in an OWL file.

The process layer of our framework is based on the process modelling workbench [2] developed in the OntoGov project[5]. It allows to visually model business processes and to annotate them with entities from ontologies. This information is leveraged for ontology-based change management – i.e. propagating changes from domain ontologies to depending processes. Process descriptions are saved in a profile and a process ontology for each process (see Figure 3). Within the process modelling workbench, there is the possibility to verify processes against the previously defined rules by invoking the inference component.

5. CONCLUSION

In this paper, we presented the Ontoprocess architecture for semantic business process management. It is implemented by a simple rule editor and a process modelling workbench that uses formal methods to verify process models against business rules. Using example data from the domain of veterinary regulation we have shown how our framework can assist process engineers by automatically identifying inconsistencies in process models.

We see two major benefits in this approach. First, the speed and efficiency of change management rises. While process engineers have to check every process in order to be sure of its compliance in standard environments, Ontoprocess helps to highlight processes that become inconsistent in the case of rule or ontology changes. Secondly, the rules can guarantee the compliance of business processes, given that they are correctly annotated. Domain and process models can be maintained by appropriate experts, thus allowing a separation of concerns. Domain models and rules may be centrally created or even “bought” from third parties, while business engineers can concentrate in managing their process models.

As a drawback, one might consider the costs for creating and maintaining the domain ontologies, rules and the annotation of services. While we think that the above mentioned advantages already compensate these costs, the usage of annotations is not limited to this scenario. The same domain models and annotations may be used for reusing and analyzing processes in the process modelling workbench and for discovery and matching of semantic web services at process runtime.

6. ACKNOWLEDGMENTS

This work was carried out as a part of the OntoGov project (OntoGov IST-2002-507237), which is partially funded by the European Commission under the IST program. It has also been partially funded by SAP Research Center CEC Karlsruhe.

7. REFERENCES

[1] B. Motik, U. Sattler, and R. Studer: Query Answering for OWL-DL with Rules, *Proc. of the 3rd Int. Semantic Web Conference (ISWC 2004)*, Hiroshima, Japan, November, 2004

[2] L. Stojanovic, D. Apostolou: Ontology-based Change Management of e-Government Services, *Web Intelligence (WI2005) Conference* - "Semantics and Orchestration of eGovernment Processes" Workshop, France, 2005

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

IkeWiki: A User-Friendly Semantic Wiki

Sebastian Schaffert, Rupert Westenthaler, and Andreas Gruber
Salzburg Research Forschungsgesellschaft
Jakob Haringer Str. 5/II, A-5020 Salzburg, Austria
http://www.salzburgresearch.at

ABSTRACT

This article describes the architecture and design principles of *IkeWiki*, a Semantic Wiki we developed at Salzburg Research. Outstanding features of IkeWiki are its support for collaborative knowledge engineering, its ease of use, its support for different levels of formalisation ranging from informal texts to formal ontologies, and its sophisticated, interactive user interface.

1. INTRODUCTION

A serious obstacle for the development of Semantic Web applications is the lack of formal ontologies and knowledge represented in other formal languages. Arguably, one of the main reasons for this is the rather high technical barrier for using Semantic Web technologies that deters many domain experts from formalising "their" knowledge.

At the same time, wiki systems are becoming more and more popular as tools for content and knowledge management. Much knowledge is nowadays available in systems like Wikipedia. Unfortunately, this vast knowledge is not accessible for machines. If a small amount of this knowledge would be formalised, wiki systems could provide improved interfaces and advanced searching and navigation facilities.

"Semantic Wikis" aim to contribute to this by combining Wiki and Semantic Web technology. Semantic Wikis exist in many different flavours, of which we only mention a few for space reasons. In this article, we present our feature-rich prototype system *IkeWiki*[1] (*ike* = *knowledge*, *wiki* = *fast*).

Semantic Wikis make the inherent structure of a wiki – given by the strong linking between pages – accessible to machines (agents, services) beyond mere navigation. Such annotations are useful for many purposes, e.g. enhanced presentation by displaying contextual information, enhanced navigation by giving easy access to relevant related information, and enhanced "semantic" search that respects the context in addition to the content.

[1]available at http://ikewiki.salzburgresearch.at

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006
.

Typing/Annotating of Links. Semantic Wikis allow to annotate links by giving them certain types. The rationale is that a link created by a user almost always carries meaning beyond mere navigation. Some semantic wikis include the annotations as part of the wiki syntax (e.g. *SeMediaWiki* [1]), while others provide a separate editor for adding annotations (e.g. *IkeWiki*).

Context-Aware Presentation. Many semantic wikis can change the way content is presented based on semantic annotations. This can include enriching pages by displaying of semantically related pages in a separate link box, displaying of information that can be derived from the underlying knowledge base (e.g. license information), or even rendering the content of a page in a different manner that is more suitable for the context (e.g. multimedia vs. text content).

Enhanced Navigation. In Semantic Wikis, link annotations describe the relation of two pages. Such information can be used to offer additional or more sophisticated navigation, e.g. in a separate "related information" box.

Semantic Search. Most semantic wikis allow a "semantic search" on the underlying knowledge base. Usually, queries are expressed in SPARQL, and allow users to ask queries like "retrieve all pieces composed by Mozart" or "retrieve all documents where the license permits derivative works".

2. DESIGN PRINCIPLES

IkeWiki has originally been developed as a tool to support knowledge workers in collaboratively formalising knowledge [2]. Its design principles are influenced by this idea:

Easy to Use, Interactive Interface. As domain experts are usually non-technical people, IkeWiki aims to be easy to use. Its interface resembles as closely as possible the familiar Wikipedia interface. Also, IkeWiki offers an interactive WYSIWYG editor (using AJAX technology to communicate with the server backend) in addition to the traditional structured text editor. The WYSIWYG editor also supports interactive typing of links and resources.

Compatibility with Wikipedia. A significant amount of "informal" knowledge is available in Wikipedia. To reuse this knowledge, IkeWiki supports the Wikipedia syntax. This allows users to import existing content from Wikipedia into IkeWiki (e.g. via simple copy and paste) and begin with semantic annotations straight away.

Compatibility with Semantic Web standards. To be able to exchange data with other applications (e.g. ontology editors, Semantic Web services, other wikis), IkeWiki is purely based on existing Semantic Web standards.

Immediate Exploitation of Annotations. An impor-

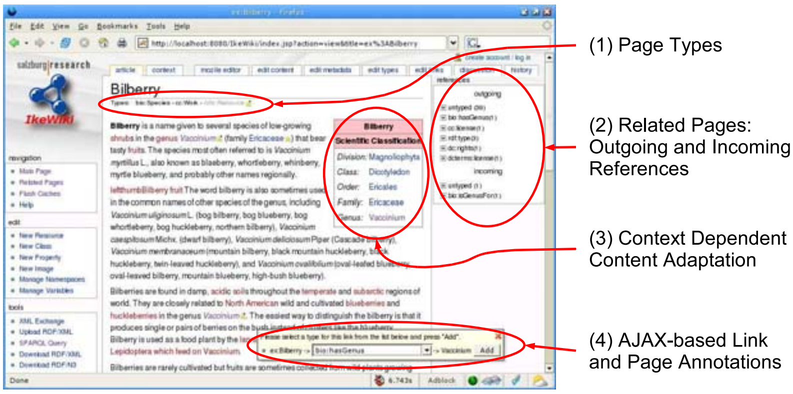


Figure 1: IkeWiki interface (cf. Section 4)

tant motivating aspect of wiki systems is that content is immediately publicly available. Similarly, IkeWiki allows immediate exploitation of semantic annotations for enhanced editing, presentation, navigation, and searching, even if the knowledge base is not yet fully formalised.

Support for Different Levels of Experience. IkeWiki is designed as a tool for collaborative knowledge engineering, where it is common that non-technical domain experts work together with experienced knowledge engineers. Therefore, IkeWiki supports all levels of experience. This means that certain advanced functionalities can be hidden from novice users but are available to experienced users.

Support for Different Levels of Formalisation. Different application areas need different levels of formalisation [2]. One of the goals of IkeWiki is thus to support formalisation of knowledge all the way from informal texts to formal ontologies. Also, this means that parts of the knowledge base might be more formalised than others, and that formal knowledge is in constant evolution.

Support for Reasoning. Unlike most other Semantic Wikis, IkeWiki supports reasoning. This allows to derive additional, implicit knowledge from the knowledge base using pre-defined or user-defined rules.

3. ARCHITECTURE

IkeWiki is implemented as a Java web application using a layered architecture. Data is stored in a Postgres database. When a resource is requested, the XML page content and related RDF data are retrieved and combined in the RenderingPipeline into an enriched XML representation. This XML representation is then either offered as interchange format for other Web services or transformed into HTML for presentation in the user's browser.

Page Store. The page store component serves to store and retrieve page content from the database, and supports versioning. Page content is represented in an XML format we call *WIF* (wiki interchange format). Basic WIF merely describes the page content and structure, but allows to add custom, application specific information.

RDF Store. The knowledge base is represented using the Jena RDF framework. Part of the RDF store is a SPARQL engine that allows for searching of the knowledge base.

Rendering Pipeline. The Rendering Pipeline combines page content with semantic annotations. Its output is a WIF document enriched by relevant semantic annotations. The "pipeline" consists of small "wiklets" that add specific pieces of information to the WIF document. Wiklets can be enabled and disabled and associated with permissions so that only selected users can see the added information.

4. INTERFACE

IkeWiki uses a purely browser-based interface (cf. Figure 1). It currently only supports the Mozilla browser family due to its standards compliance and free availability.

Page View. Figure 1 shows a sample article (copied from Wikipedia) about the "Bilberry" as rendered in IkeWiki. Type information is given below the page title (1). Links to (semantically) related pages are displayed in a separate "references box" on the right hand side (2). The taxonomy box (3) showing the biological classification of the described plant is automatically generated from existing semantic annotations (i.e. Bilberry *hasGenus* Vaccinium) and is an example for context adaptation. Finally, (4) shows interactive typing of links using AJAX technology.

Content Editor. The content editor is available in two flavours: as a traditional structured text editor and as a WYSIWYG editor. The structured text editor is aimed at expert users that are familiar with other wiki systems, and allows to directly copy content from Wikipedia. The WYSIWYG editor is aimed at novice users creating new content and interacts closely with the server backend: links are automatically recognised and verified, and links can be annotated directly in the editor (Fig. 1, item (4)).

Semantic Annotations Editor. Semantic annotations are separated into three editors: the *metadata editor* allows to fill in textual metadata related to a page (like Dublin Core metadata or RDF comments). The *type editor* allows to associate one or more of the types available in the system with a page. The *link editor* allows to annotate outgoing and incoming links.

5. REFERENCES

[1] M. Krötsch, D. Vrandečić, and M. Völkel. Wikipedia and the Semantic Web - The Missing Links. In *Proceedings of the WikiMania2005*, 2005.

[2] S. Schaffert, A. Gruber, and R. Westenthaler. A Semantic Wiki for Collaborative Knowledge Formation. In *Semantics 2005*, Vienna, Austria, November 2005.

Demo: Publishing Desktop Data with semiBlog*

Knud Möller
Digital Enterprise Research Institute
National University of Ireland
Galway
knud.moeller@deri.org

ABSTRACT

This demo proposal briefly presents the *semiBlog* editing platform for Semantic Blogging. semiBlog allows a blog author to reference objects from other desktop applications — addressbook entries, events, publications, etc. — and annotate blog posts with these objects. The annotations are realized as RDF resources generated from the referenced desktop objects.

1. INTRODUCTION

Blogging, as a subset of the web as a whole, can benefit greatly from the addition of semantic metadata. The result — which we will call *Semantic Blogging* — provides improved capabilities with respect to search, connectivity and browsing compared to current blogging technology. Moreover, Semantic Blogging will allow new ways of convenient data exchange between the actors within the blogosphere — blog authors and blog users alike.

2. SEMIBLOG

Semantic Blogging has been identified as a topic and discussed in a number of recent publications (especially [2], [1], [3] and [7]). The semiBlog application[1] (previously presented in [5] and [6], as well as in the research track of ESWC2006 [4]) takes up these ideas on semantic blogging, adds to them integration with a user's personal desktop (i.e. the data on their computer), and addresses usability questions, in order to maximize a user's incentive to use the technology.

The basic assumption behind semiBlog is that a user often wishes to blog about topics or things for which they already have formal data available in some form. Such topics or things might e.g. be people, events like conferences or meetings, publications or music. For all these things, metadata will often already exist in a user's electronic addressbook, calendaring application, bibliographic database or mp3 collection — in short, somewhere on the desktop. semiBlog uses a plugin architecture (cf. Fig. 1) to wrap the various desktop based data sources. Each plugin is responsible for data of a specific type, and contains functionality to resolve references to relevant (structured or semi-structured) data

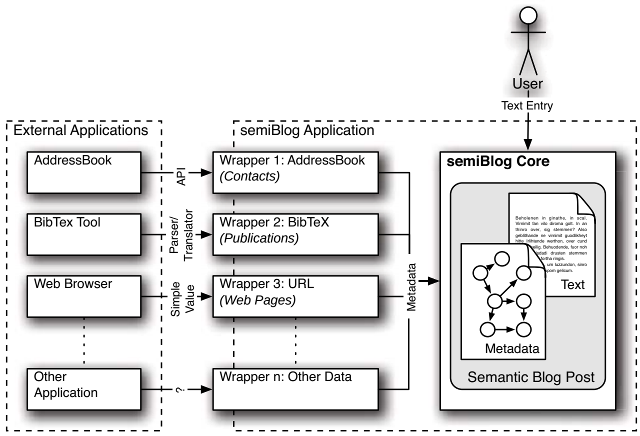


Figure 1: Creating RDF metadata from desktop applications

objects, as well as translate them into a common, Semantic Web conform data model: RDF. Through techniques such as drag and drop from different desktop applications, a user can annotate a blog post with metadata about the post's topic. Fig. 2 exemplifies this: here, the user has authored a blog post about an ESWC paper, and subsequently annotated the post with metadata taken from their electronic addressbook (the authors of the paper) and a webbrowser (the ESWC2006 web site).

Connecting a blog post to existing desktop data provides a number of benefits: *(i) reuse of data* — once a user has entered metadata into any of the applications they use on a day-to-day basis, there is no need to enter it again when annotating a blog post. This makes a desktop based application like semiBlog different from web based approaches, which offer HTML forms to add metadata. *(ii) reuse of functionality* — specialized external applications like an electronic addressbook or calendar are usually very good at dealing with the kind of data they are designed for. It is therefore beneficial to reuse this expertise, instead of reimplementing inferior methods of metadata creation. *(iii) always up-to-date* — semiBlog links to desktop objects instead of duplicating them in RDF. This means that, should a user update data in an external application, this update will automatically be reflected in their blog.

Just as importing data from other applications is implemented using a plugin architecture, functionality for pub-

*This material is based upon works supported by the Science Foundation Ireland under Grant No. SFI/02/CE1/I131.
[1] http://semiblog.semanticweb.org

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

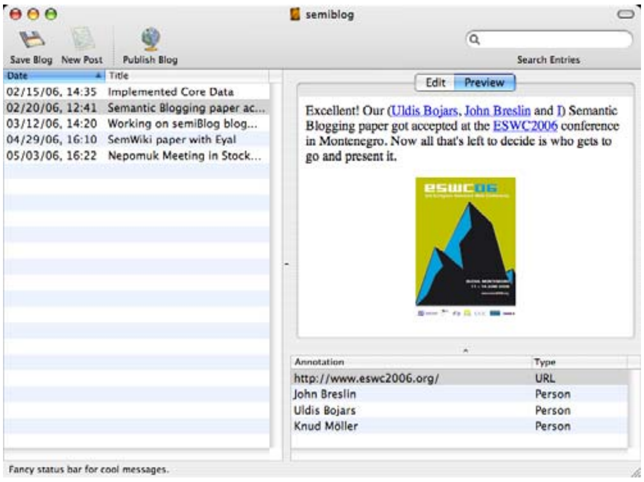


Figure 2: The semiBlog editing environment

lishing the blog in various ways is implemented as plugins to the core application. We currently offer a publishing plugin which generates individual RDF files for each annotation object and uploads them to a WordPress[2] server, using the Metaweblog API[3] through XML-RPC. The actual blog posts will then contain links to these RDF files. However, an implementation which uses an external RDF repository such as YARS[4] or Sesame[5] is equally possible. Also, publication based on the Structured Initiative[6] could be implemented as a plugin.

semiBlog is implemented as a desktop-based application. Since tight integration with other desktop applications was a goal when developing the software, we decided to realize semiBlog in a platform-specific way. Currently, semiBlog is therefore only available as a Mac OS X application.

3. DEMONSTRATION

For the demonstration, we will show how a user authors a simple blog post within semiBlog, consisting of text and pictures. We will then show how to annotate the post with data from various external applications using drag and drop (currently an addressbook and web browsers are supported; we will have additional plugins available for the conference), publish the post to a blogging platform such as Wordpress and show the resulting post on the web. Additionally, we will show how changes to data in the external applications are reflected in the blog. Apart from a guided demonstration by the authors, conference attendees will also have the opportunity to try out the software and play with it themselves.

4. REFERENCES

[1] S. Cayzer. Semantic Blogging and Decentralized Knowledge Management. *Communications of the ACM*, 47(12):47–52, December 2004.

[2] S. Cayzer. Semantic Blogging: Spreading the Semantic Web Meme. In *XML Europe 2004, Amsterdam, Netherlands, Proceedings*, April 2004.

[3] D. R. Karger and D. Quan. What Would It Mean to Blog on the Semantic Web? In S. A. McIlraith, D. Plexousakis, and F. van Harmelen, editors, *Third International Semantic Web Conference (ISWC2004), Hiroshima, Japan, Proceedings*, pages 214–228. Springer, November 2004.

[4] K. Möller, U. Bojārs, and J. G. Breslin. Using Semantics to Enhance the Blogging Experience. In *The third European Semantic Web Conference, Budva, Montenegro*, June 2006.

[5] K. Möller, J. G. Breslin, and S. Decker. semiBlog - Semantic Publishing of Desktop Data. In *14th Conference on Information Systems Development (ISD2005), Proceedings*, Karlstad, Sweden, August 2005.

[6] K. Möller and S. Decker. Harvesting Desktop Data for Semantic Blogging. In *1st Workshop on the Semantic Desktop at ISWC2005, Galway, Ireland, Proceedings*, pages 79–91, November 2005.

[7] H. Takeda and I. Ohmukai. Semblog Project. In *Activities on Semantic Web Technologies in Japan, A WWW2005 Workshop*, 2005.

[2] `http://wordpress.org/`
[3] `http://www.xmlrpc.com/metaWeblogApi`
[4] `http://sw.deri.org/2004/06/yars/`
[5] `http://www.openrdf.org/`
[6] `http://structuredblogging.org/`

SGSDesigner, the ODESGS Environment User Interface

Asunción Gómez-Pérez
Ontology Engineering Group
Campus de Montegancedo s/n,
28660 Boadilla del Monte, Madrid. Spain.
(+34) 913367439
asun@fi.upm.es

Rafael González-Cabero
Ontology Engineering Group
Campus de Montegancedo s/n,
28660 Boadilla del Monte, Madrid. Spain
(+34) 913367439
rgonza@fi.upm.es

ABSTRACT
In this demo, we will show SGSDesigner, the ODESGS Environment user interface. ODESGS Environment (the realization of the ODESGS Framework [1]) is an environment for supporting both a) the annotation of pre-existing Grid Services (GSs) and b) the design of new complex Semantic Grid Services (SGSs) in a (semi) automatic way. In the demo we will focus in the annotation of a WSRF GS, using the annotation process proposed by the ODESGS Framework.

Categories and Subject Descriptors
D.2.2 [Software Engineering]: Design Tools and Techniques-*user interfaces*

General Terms
Design, Theory.

Keywords
Semantic Grid Services, Problem-Solving Methods.

1. INTRODUCTION
GSs are defined as a network-enabled entity that offers users some capabilities. In order to make them computer-interpretable, user-apparent and agent-ready, we annotate them with formal and explicit semantics.

Our proposal for annotating GS (and designing new complex SGS from these annotated ones) is the ODESGS Framework [1]. It uses ontologies and Problem-Solving Methods (PSMs) to describe the features of GS operations (a PSM is defined as a domain-independent and knowledge-level specification of the problem solving behavior that can be used to solve a class of problems [2]).

The process of annotating a GS in the ODESGS Framework (what we will show in this demonstration) starts with the creation of the knowledge level description (step 1 in Figure 1). Once this GS knowledge level description is created (the GS can be optionally checked, step 2), it can be automatically translated into different representational instantiations (the last step in Figure 1). The knowledge and representational level descriptions are both attached to the GS by means of Semantic Bindings [3](S-Bindings), allowing thus that a single GS may have multiple descriptions, which could be expressed in different languages and formalisms.

As we have already stated, ODESGS Environment realizes the ODESGS Framework. Therefore the front-end of the ODESGS Enviroment should be a graphical user interface that gives support to all these steps of the annotation process shown in Figure 1. This interface, on the one hand, should allow the easy creation of these knowledge level specifications (i.e. should facilitate knowledge entities creation), and on the other hand, should aid the user to attach pre-existing GSs to these descriptions (i.e. should facilitate S-Bindings creation). Finally, it must be able to communicate with different translators. Therefore, SGSDesigner current main features are:

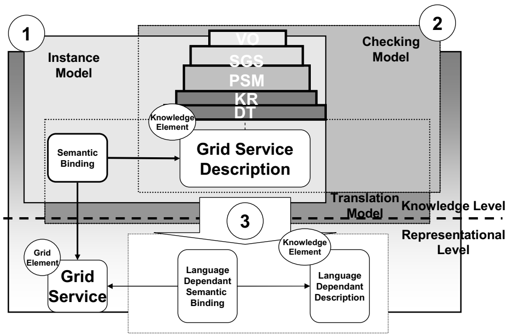


Figure 1 ODESGS Framework Annotation Process.

- ***Knowledge level descriptions are made in a graphical fashion***. The user is not aware of the formalisms used to represent the service; the whole design of the PSMs and other diagrams is carried out just by drawing or dragging and dropping knowledge components.
- ***Semantic markup export capable***. Once the knowledge level description and the S-Bindings are defined, an RDF(S) representation of them can be automatically generated. The service model that is used is being developed in the Ontogrid Project[1].
- ***Multiple and heterogeneous ontologies handling***. The editor can use OWL and RDF(S) ontologies stored in files or ontologies available in any instance of the WebODE ontology workbench.
- ***WSRF Compliant***. Currently, SGSDesigner is able to annotate WSRF[2] GSs, as it will be described later.

2. SGSDesigner
The SGSDesigner design has been inspired in the classical representation of PSMs. It includes hierarchical trees of tasks (abstract domain independent operations) and methods (abstract domain independent reasoning processes); input/output interaction diagrams of the tasks; diagrams about how the sub-tasks that compose a method are orchestrated; and data flows that describe data exchange of the sub-tasks. Let us describe each of the elements of this user interface.

---

[1] http://www.ontogrid.net

[2] http://www.globus.org/wsrf/specs/ws-wsrf.pdf

Workspaces are the main components of SGSDesigner. They are the concept of projects in software development tools. They contain the definitions of knowledge components and provide all the mechanisms and diagrams for both defining and storing them. Therefore, before we start working with SGSDesigner for creating the description of the service, we will select, at least, one workspace to use.

Workspaces have two general areas: trees and views (they are identified in Figure 2). Let's describe their intended use in detail.

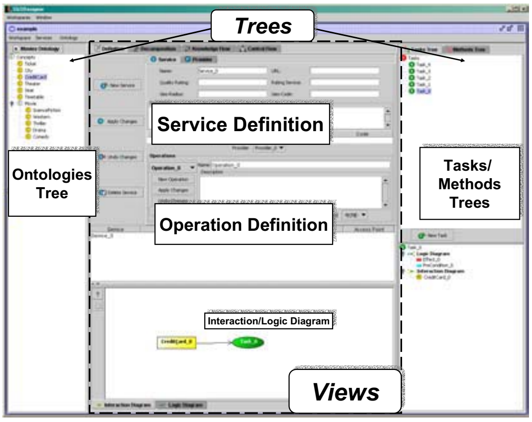


Figure 2 SGSDesigner Workspace.

Trees show the hierarchy of the knowledge components, such as tasks, methods, and ontologies elements.

- ***Ontologies trees.*** These trees show the concepts and attributes of the ontology (or ontologies) used to specify all the knowledge components. They can import the concepts and attributes that will be shown in the ontology tree. Then, the user could drag the icons representing a concept/attribute and drop them into the diagrams that enable the specification of the input/output roles of the tasks and methods.
- ***Tasks(Methods) tree.*** This tree allows users to create the tasks (methods) associated to the functional features of the operations. Once the tasks (methods) have been created, the user can drag, from this tree, the node representing a task (method) and drop it into the diagrams.

Views allow users to specify all the features of the knowledge components that describe a service, and they are represented as tabs:

- ***Definition View*** is used to specify the non-functional features of a service of the service (e.g., name, description, URL, providers, etc.) and the functional properties of each of its operation. Each of these operations is described by associating it to a task (what the operation does) and to a method (how the method performs this task). The operation task is defined by means of two diagrams: the ***Interaction Diagram,*** which specifies the type of the inputs/outputs of the task; and the ***Logic Diagram***, which contains the pre/post-conditions, assumptions and effects of the task. They are shown as colored rectangles.

  Methods are described by these diagrams plus the Knowledge/Control Flow Views.
- ***Decomposition View.*** This tree-like view allows users to specify the hierarchy of tasks and methods. Composite methods are decomposed into sub-tasks, which will be solved by other methods, and so forth. The leaves of a decomposition diagram are atomic methods.
- ***Knowledge Flow View.*** This view defines the data flow and the relationship between the inputs/outputs of the sub-tasks of a composite method.
- ***Control Flow View.*** This view describes the control flow of composite methods. The elements of this view are the sub-tasks of the method plus some workflow constructions (e.g. if-then, while-until, split and join, etc.)

In the demo, as we will annotate a pre-existing WSRF GS, we will create a task and its describing atomic method. Nevertheless, we will also briefly show how to use all these views to create a complex SGS, creating thus several task and several composite methods.

2.1 WSRF Annotation

Once the knowledge level description of the SGS has been created with the aforementioned views and trees, we will use the WSRF Import Wizard. It will guide us through several steps, which will finally associate each of the operations (and its inputs and outputs) of the GS to the knowledge level description of these operations (and the input and output roles of its associated task as Figure 3).

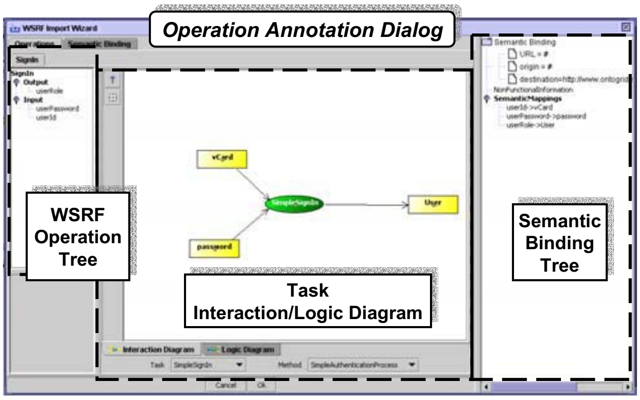


Figure 3 WSRF Service Annotation.

Finally, we will obtain the RDF instance of both the GS description and/or the S-Binding.

3. ACKNOWLEDGMENTS

This work has been partially financed by the Ontogrid Project (FP6-511513) and by a grant provided by the Comunidad Autónoma de Madrid (Autonomous Community of Madrid).

4. REFERENCES

[1] C. Goble, A. Gómez-Pérez, R. González-Cabero, M. Pérez-Hernández (2005) ODESGS Framework, Knowledge-based annotation and design of Grid Services, ICSOC 2005 341-352

[2] E.Motta (1999): Reusable Components for Knowledge Modeling, IOS Press

[3] I. Kotsiopoulos, P. Alper, O. Corcho, S. Bechhofer, C. Goble, D. Kuo, P. Missier Ontogrid Deliverable D1.2 Specification of a Semantic Grid Architecture October 7th, 2005

ESWC 2006 Demo: DBin – enabling SW P2P communities

G.Tummarello, C.Morbidoni, M. Nucci, F.Piazza, P.Puliti - Università Politecnica delle Marche, Italy

http://semedia.deit.univpm.it – http://www.dbin.org

ABSTRACT

DBin is a general purpose, integrated, visually rich, open source, multiplatform Semantic Web that can be demonstrated and delivered to the end user today. With DBin, thanks to an integrated P2P engine, users can cooperatively annotate any domain of interest (under the methaphor of "group"). As individual users collect RDF from P2P groups and from any other sources, they are able to search and browse merged information in a maximally fast, rich and personalized way. DBin accommodates a number of modules to deal with specific issues ranging from visualization to trust.

1. Introduction

DBin is a user centered knowledge management platform revolving around a local, personal, Semantic Web Database. Content is inserted in this database in a number of ways:

- by a novel P2P Semantic Web algorithm (RDFGrowth) therefore fed from other DBin installations
- by specific modules integrating the content of the local machine (desktop integration).
- Explicitly by the users (which therefore contribute to the P2P knowledge)
- By the inclusion of external data sources or RDF graphs

All the knowledge stored in DBin is expressed using the languages defined in the Semantic Web initiative (RDF, RDFS) but the user doesn't necessarily have to be aware of this as the rich user interface will make it unnecessary to see or understand the basic information blocks.

2.Use scenario

A typical use of DBin might be similar to that of popular file sharing programs, the purpose however being completely different. While usual P2P applications "grow" the local availability of data, DBin grows RDF knowledge.

Once a user has selected the topic of interest and has connected to a semantic web P2P group, RDF annotations just start flowing in and out "piece by piece" in a scalable fashion. Such operations are clearly topic agnostic, but for the sake of the demonstration lets take an example of possible use of DBin by a Semantic Web researcher.

For example, a user who expresses interest in a particular topic and related papers (say "Semantic Web P2P") will keep a DBin open (possibly minimized) connected with a related P2P knowledge exchange group. He will then be able to review from time to time new pieces of relevant "information" that DBin collects from other participants. Such information might be pure metadata annotations (e.g. "the deadline for on-topic conference X has been set to Y") but also advanced annotations pointing at rich data posted on the web (pictures, documents, long texts, etc..). He could then reply or further annotate each of this incoming pieces of info either for his personal use or for public knowledge. If such replies include attachment data, DBin automatically takes care of the needed web publishing. At database level, all this information is coherently stored as RDF. At the user level however, the common operations and views are grouped in domain specific user interfaces, which in DBin are called "Brainlets".

3.Brainlets

Brainlets can be though of "configuration packages" preparing DBin to operate on a specific domain (e.g. Wine lovers, Italian Opera fans etc..). Given that Brainlet include customized user interface, the user might perceive them as full "*domain applications run inside DBin*" which can be installed as plug-ins and are suggested as soon as the user tries to enter a P2P group associated with the Brainlet itself. The message the user sees is similar to "The group you're trying to enter contains information which is best experienced with the X Brainlet, please visit page Y and install it". Continuing without said Brainlet is possible, but the interface wont be optimal for the given domain. In short Brainlets define settings for:

- The ontologies to be used for annotations in the domain
- A general GUI layout;. which components to visualize and how they are cascaded in terms of selection/reaction
- Templates for domain specific "annotations", e.g., a "Movie" brainlet might have a "review" template that users fill.
- Templates for readily available "pre cooked" domain queries.
- Templates for wizards which guide the user when inserting new domain elements (to avoid duplicated URIs etc)
- A suggested trust model and information filtering rules for the domain. e.g. Public keys of well known "founding members" or authorities,
- Basic RDF knowledge package for the domain

Creating Brainlets doesn't require programming skills, as it is just a matter of knowledge engineering (e.g. selecting the appropriate Ontologies) and editing of XML configuration files.

4.RDFGrowth: a scalable P2P engine based on the "minimum commitment" principle

The RDFGrowth algorithm powers DBin ability to collect RDF metadata from other peers with common interests. Previous projects, have explored P2P interactions among peers that rely on each other to forward query requests, collecting and returning results [3]. In contrast, RDFGrowth is designed to operate in a particularly "greedy" and

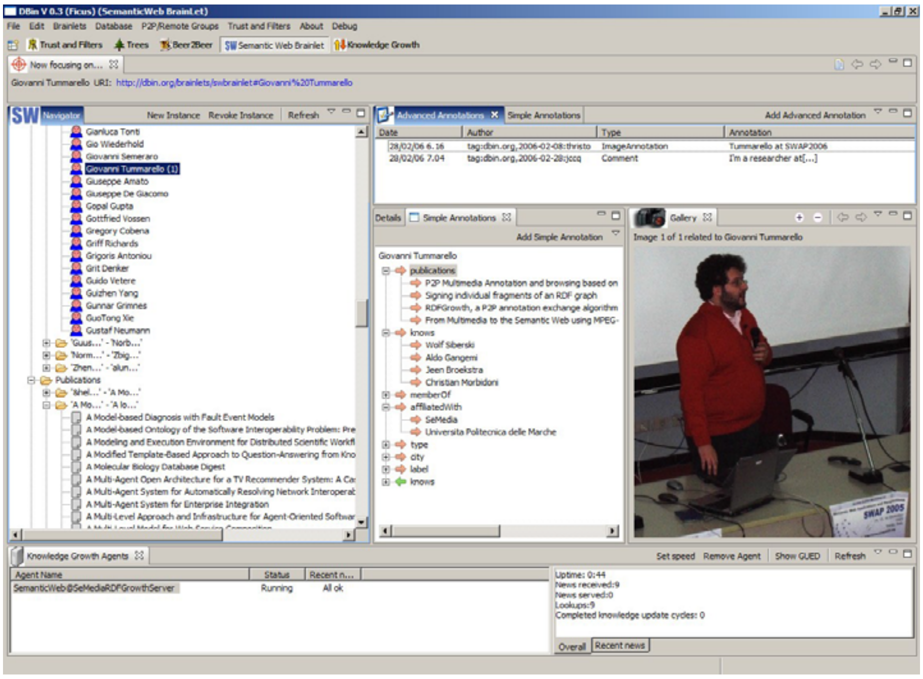


Figure 1 A screen shot of the SemanticWeb research Brainlet running. The principal "views" are: an ontology (and instances) browsing Navigator, a set of "Annotation" views and others related to searching, browsing, filtering etc. Kickstarting data is delivered insider the brainlet itself and has been adapted from that kindly made available by the Flink project [4].

uncommitted scenario where cooperation between peers is minimal. By this we mean that while peers are willing to provide some external service, the commitment should be minimal and in a "best effort" fashion. To obtain this, RDFGrowth follows a peculiar philosophy: minimum external burden.

- Given that a complex graph query could simply hog any machine, we assumed that individual peers would not, in general, be willing to answer arbitrary external queries. Any single peer would, if at all, answer just very basic ones. RDFGrowth only requires peers to answer very simple queries: basically the "RDF Surroundings" or blank node closure of the triples surrounding a specific URI. This type of query is not only very fast to execute but can also be cached very effectively.
- No "active information hunt" such as query routing, replication, collecting and merging is done. Such operations would require peers to do work on behalf of others that is again allowing peers to cause a potentially large external burden.

So, instead of querying around, in DBin a user browses only on a local and potentially very large metadata database, while the RDFGrowth algorithm "keeps it alive" by updating it in a sustainable, "best effort" fashion. A complete discussion is outside the scope of this introduction to the Demo, those interested can refer to [1] and other papers available from the DBin web site. As a result, keeping DBin open and connected to P2P groups with moderate traffic requires absolutely minimal network and computational resources.

5.Trust and URI Bridge Component

Due to the open nature of the P2P model (which can however be restricted to be used within organization or intranet), DBin also implements an RDF digital signature infrastructure that can be used by end users to perform custom trust based information filtering as well as signing annotations to be inserted in the system. For more details about the trust theory and infrastructure, see [2].

6.Conclusions and related works

DBin is an end user/power user centered application which provides an undoubtedly simplified, yet novel and exciting, all round and integrated Semantic Web experience. To the best of our knowledge there are no other projects which face the "all round" user scenario. Aspects of DBin capabilities can be directly compared with [5][6][7]. DBin is an Open Source project (GPL). Further documentation and compiled executables can be downloaded at http://dbin.org.

7.References

[1] G. Tummarello, C. Morbidoni, J. Petersson, Paolo Puliti, Francesco Piazza, "RDFGrowth, a P2P annotation exchange algorithm for scalable Semantic Web applications", 2004

[2] G. Tummarello, C. Morbidoni, P. Puliti, F. Piazza "Signing individual fragments of an RDF graph", WWW2005, poster track, Chiba

[3] W.Nejdl, B.Wolf "EDUTELLA: A P2P Networking Infrastructure Based on RDF" www2002 Honolulu

[4] Flink Project - http://prauw.cs.vu.nl:8080/flink/

[5] David Huynh, Stefano Mazzocchi, and David Karger. Piggy Bank: Experience the Semantic Web Inside Your Web Browser. International Semantic Web Conference (ISWC) 2005.

[6] P. Haase, J.Broekstra, M Ehrig, M. Menken, P. Mika, M. Plechawski. P. Pyszlak, B. Schnizler, R. Siebes, S. Staab, C. Tempich "Bibster --- A Semantics-Based Bibliographic Peer-to-Peer System", ISWC2004

[7] Quan, Karger, "How to make a Semantic Web Browser" WWW2004

Posters

Ontologies, Questionnaires and (Mining) Tabular Data

Vojtěch Svátek
Department of Information and Knowledge Engineering, University of Economics, Prague
W. Churchill Sq. 4, 13067 Praha 3, Czech Republic
svatek@vse.cz

ABSTRACT

Questionnaires are an interesting source for ontology design, especially in connection with KDD applications. Two case studies from different domains are presented.

Keywords

Ontological engineering, knowledge discovery from databases

1. INTRODUCTION

Ontology-based knowledge discovery techniques are mostly applied on loosely structured textual or multimedial data. However, a significant portion of the information wealth of the mankind is latently present in structured databases from which it can be extracted by means of KDD (knowledge discovery from databases) techniques. In the rest of this text, we will refer to this kind of data as to *tabular*. Recently, the potential role of ontologies as prior knowledge in the KDD process has been discussed in the framework of workshops on 'Knowledge Discovery and Ontologies' [1, 2].

In this paper we also take into account a third resource, which seems to have genuine connection both to ontologies and databases, namely, *questionnaires* that are often used to collect tabular data. Typical interactions among the three types of resources are depicted inFull lines correspond to creation of resources, while dashed ones correspond to provision of additional information.

The questionnaire has twofold impact on the database: its structure is transferred into that of the data, and the textual labels clarify the semantics of the fields to humans. The texts in the questionnaire can, however, also serve as resource of ontology entities: classes, relations as well as instances (to say, values of closed questions). Similarly, the data tables (namely, values of fields corresponding to open questions) can serve as resource of instances for the ontology. Finally, the ontology can impact the analysis of tabular data in several ways: to (semi-)automatically focus the mining process, provide interpretations of discovered results, allow to expose the results on the semantic web and the like [3].

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006
.

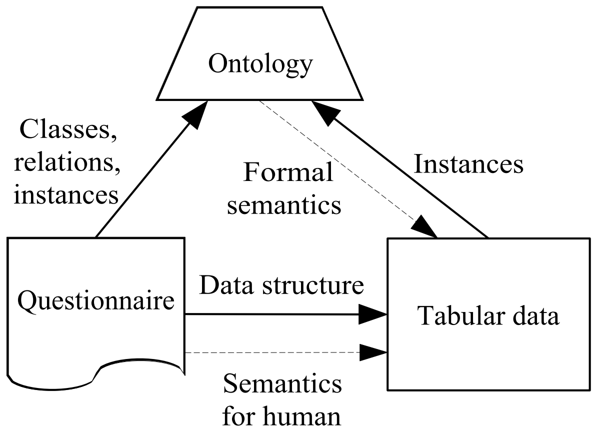


Figure 1: Interactions among the three resources

In section 2 we briefly discuss the characteristics of questionnaires as input for ontology design. In section 3 we report on case studies undertaken in two different projects. Finally, section 4 gives a summary and future plans.

2. QUESTIONNAIRES AS RESOURCES IN ONTOLOGY DESIGN

In ontology engineering methodologies, pre-existing questionnaires are hardly considered as stand-alone resources; Obviously, they are not always available and their scope is typically too narrow for a large-scale domain ontology. However, when incrementally building an ontology covering the portion of domain relevant to a given collection of tabular data, questionnaires may be quite useful:

- their *small extent* and *richness* in domain terms makes them amenable to manual processing with no or just light-weighted suppport by automated NLP
- most terms are relevant not only wrt. the domain but also wrt. the *applications* that would process the data; this alleviates the well-known 'hugeness' problem of ontology design
- the structure of questions (and answer options) may provide some cues to resulting *ontology structures*
- the *mapping* between data fields and ontology concepts is immediately available for future use.

In the following, we illustrate these advantages on two case studies.

3. CASE STUDIES

3.1 Social Reality Ontology

One of goals recently formulated in the *LISp-Miner* project[1] was to exploit domain ontologies to enhance the KDD (in particular, association mining) process and its results. For one of initial experiments, a dataset was chosen that contained information from more than 3000 respondents concerning their attitude towards various events of social and political life in the city of Prague (in 2004, the year of the country entering the EU). As there was no suitable domain ontology at hand, a new one was manually designed in a bottom-up manner, taking the comprehensive *questionnaire* (with about 50 questions) used in the poll, as starting point. Terms from the questionnaire were upgraded to concepts, relations and instances for the ontology, while keeping the mapping between the data fields and ontology entities. Only a few more entities were later added, in cooperation with a domain expert, to achieve connectivity of the whole model. The resulting (OWL) ontology contained about 100 classes, 40 relations and 50 instances. Eventually, the database was analysed using the *LISp-Miner* [4] tool, and some of the discovered associations were endowed with potential 'explanation paths' from the ontology [7].

Several explanation paths were characterised as interesting and to some degree 'plausible' by the domain expert. See for example the path "KSCM $\in$ Political_party *isRepresentedIn* Administrative_body $\sqsupseteq$ City_council *carriesOutAction* Economic_action *hasImpactOn* Social_phenomenon $\ni$ bad_living_standard". It explains the empirical association between the question "Do you expect that the standard of living of most people in the country will grow?" with answer 'certainly not', and the question "Which among the parties represented in the city council has a programme that is most beneficial for Prague?" with 'KSČM' (the Czech Communist Party) as answer, (roughly) as "KSČM party is represented in the city council, which can carry out an economic action, which may have some impact on the phenomenon of bad living standard".

3.2 Conference Organisation Ontologies

The *OntoFarm* project [6] aims at independent development of multiple ontologies of the same domain—that of *conference organisation*—thus providing a benchmark collection for ontology-processing techniques such as automated alignment, distributed reasoning or discovery of implicit design patterns. Most ontologies were designed based on human or automated analysis of either conference-support *software tools* (incl. documentation), *websites* of concrete conference series, or *insider info* on organising a conference. Eventually, we decided to include a fourth resource (obviously covering a fragment of the whole domain only), namely, the *review forms* as special kind of questionnaires. This fragment should address tasks such as identification of gaps and redundancies in the coverage of review forms or identification of potential inconsistencies in the reviews themselves.

The model was first created based on a single review form (for conference A) and then updated based on another form (for conference B), its size eventually amounted to approx. 30 classes and 20 relations. The important finding in this small study was that the structural aspects of the forms bring to light different modelling choices. For example, while form A only suggests to categorise the paper as either *theoretical* or *applicative* (which naturally leads to subclassing of class Paper), form B explicitly introduces the notion of *domain* in which the approach is applied (to be most faithfully modelled using a property such as *appliedIn*). Analogously, while form B only introduces the ordinal ranking according to *originality* (to be probably modelled as property with enumerated value set, cf. [5]), form A explicitly asks about prior papers with *similar content* (which calls for property expressing the 'similarity' relationship).

[1] `http://lispminer.vse.cz`

4. CONCLUSIONS AND FUTURE WORK

Based on two case studies, we discussed the role of questionnaires as input for ontology design, aiming at analysis of tabular data with the help of such ontology.

While the described experiments in questionnaire-based ontology design were manual, we are considering to implement a supporting tool. Such tool would definitely be interactive, would probably rely upon a POS tagger (as most ontology learning tools do), but would also include some kind of field detector (as form analysis tools do) in order to capture e.g. values for closed questions.

5. ACKNOWLEDGMENTS

The research is partially supported by the CSF grant no.201/05/0325 "New methods and tools for knowledge discovery in databases" and by the IGA VSE grant no.26/05 "Methods and tools for ontological engineering"; the OntoFarm project is also partially supported by the Knowledge Web Network of Excellence (IST FP6-507482).

6. REFERENCES

[1] M. Ackermann, B. Berendt, M. Grobelnik, V. Svátek (eds.) ECML/PKDD 2005 Workshop on Knowledge Discovery and Ontologies (KDO-05), Porto 2005.

[2] P. Buitelaar, J. Franke, M. Grobelnik, G. Paass, V. Svátek (eds.) ECML/PKDD 2004 Workshop on Knowledge Discovery and Ontologies (KDO-04), Pisa 2004.

[3] H. Češpivová, J. Rauch, V. Svátek, M. Kejkula, M. Tomečková. Roles of Medical Ontology in Association Mining CRISP-DM Cycle. In: [2].

[4] J. Rauch, M. Šimůnek. An Alternative Approach to Mining Association Rules. In: Lin, T. Y., Ohsuga, S., Liau, C. J., and Tsumoto, S. (eds.), Data Mining: Foundations, Methods, and Applications, Springer-Verlag, 2005, pp. 219–238

[5] A. Rector (ed.) Representing Specified Values in OWL: "value partitions" and "value sets". W3C Working Group Note, 17 May 2005, online at `http://www.w3.org/TR/swbp-specified-values/`.

[6] O. Šváb, V. Svátek, P. Berka, D. Rak, P. Tomášek. OntoFarm: Towards an Experimental Collection of Parallel Ontologies. In: Poster Track of ISWC 2005, Galway.

[7] V. Svátek, J. Rauch, M. Flek. Ontology-Based Explanation of Discovered Associations in the Domain of Social Reality. In: [1].

Applying Semantic Web Services to Virtual Travel Agency Case Study

Maciej Zaremba
Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway, Ireland
maciej.zaremba@deri.org

Matthew Moran
Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway, Ireland
matthew.moran@deri.org

Thomas Haselwanter
Digital Enterprise Research Institute (DERI)
University of Innsbruck, Austria
thomas.haselwanter@deri.org

ABSTRACT

Online travel agencies and services provide a straightforward means for travelers to manage and book their itineraries from the convenience of their own homes or offices. As online traveling services becomes more sophisticated, it becomes increasingly possible to avoid visiting brick and mortar travel agencies even for more complex traveling needs. However, clients often must visit a multitude of travel related web sites in order to check recent developments in prices, convenience of connections, synchronize flights with airport transport etc. To overcome these limitation the Virtual Travel Agency (VTA) case study proposes applying Semantic Web services and the Web Service Modelling Execution Environment (WSMX).

Categories and Subject Descriptors

D.2.11 [**Software Architectures**]: languages, domain specific architectures, patterns

Keywords

Semantic Web Services, Service Oriented Architecture, Case study, Virtual Travel Agency (VTA)

1. INTRODUCTION

This paper describes a prototype of a VTA application demonstrating how the application of Semantic Web and Semantic Web service technology makes it possible for individual customers to organize and book their itineraries. The application allows users to impose various requirement on particular steps of their journey such as flight booking, organizing airport shuttle or train and hotel reservation. User might impose restrictions on the price of the tickets, time frame between the flight and the train or shuttle, hotel location within his destination place, etc. Services are tailored on-the-fly by the web application and place no cost burden on the travel agency. To create such tailored services through traditional software design is time consuming and likely to be uneconomic.

In our VTA case study we use WSMX[1] as a run-time environment for Semantic Web services. Enhancing Web Services with semantic descriptions provides a foundation for their semi-automatic discovery, composition, invocation and interoperation enabling seamless interactions between them [2] and keeping human interaction to a minimum. Research on WSMX aims to assess the viability of WSMO[2] framework and to provide a reference implementation of the system. WSMO constituents such as Goals, Mediators, Ontologies and Web services are expressed in WSML[3]. WSMX is composed of loosely-coupled components that carry out various tasks related to WSMO. Some of the main components of WSMX are Service Discovery, Data Mediation, Process Mediation, Service Selection, and Communication Manager.

[1]Web Services Execution Environment (WSMX) - http://www.wsmx.org

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

2. PROTOTYPE DESCRIPTION

The prototype executes a VTA case study leveraging Semantic Web services technology in terms of WSMO framework. This approach has several advantages over purely syntactical XML-based interaction solutions including its ability to express partners' complex behaviours in terms of WSMO Choreography [3], mediation between data and process representations, and dynamic discovery. In a nutshell, semantic descriptions provide a foundation for logic reasoning about service description and behaviour. This section presents the necessary steps to set up semantically-enabled interaction, the description of the developed prototype, and benefits stemming from the semantic integration.

In order to semantically integrate a client with the VTA provider's Web services, both the capability and the behaviour of the interacting parties have to be semantically described. The client expresses the requested functionality and expected behaviour (choreography) in terms of WSMO Goal, while the capability and choreography offered by the provider is described as a WSMO Web service.

The following preliminary steps have to be taken:

- **Creating WSMO Goals**. The requirements and behaviour of the client has to expressed as WSMO Goal. In VTA case, Goals are based on a template approach where the Goal structure is defined but actual input values can be provided during the run-time by the client. The web application provides forms where user can specify his requirements and input values.

[2]Web Services Modeling Ontology (WSMO) - http://www.wsmo.org

[3]Web Services Modeling Language (WSML) - http://www.wsmo.org/TR/d16/d16.1/v0.21

- **Creating WSMO Web service.** Provider's Web services has to be semantically described, which includes lifting arbitrary XML messages to the semantic level by the ontology conceptualization and describing message exchange patterns (choreographies) using the Ontologized Abstract State Machines formalism of WSMO Choreography.

- **WSML grounding to WSDL.** Bidirectional mappings between XML and WSML have to be provided.

- **Ontology mapping.** Since it is likely that interacting partners use different ontologies it is necessary to provide appropriate bidirectional mappings. WSMX takes a semi-automatic approach to this problem. Mappings between the ontologies are created during design-time by using a Data Mediation Mapping tool. This tool gives a hint of the most likely mappings by analyzing both naming convention and structure of concepts. The human's role is to ensure accuracy of these mappings and to adjust them if necessary.

Figure 1 presents this VTA scenario. The client communicates with the VTA portal via the HTTPS protocol, which provides a secure communication channel. The VTA portal allows the itineraries goals to be expressed using web forms for which appropriate WSMO Goal templates are populated with the actual values and conditions. Once a WSMO Goal with actual values is created it can be sent to WSMX.

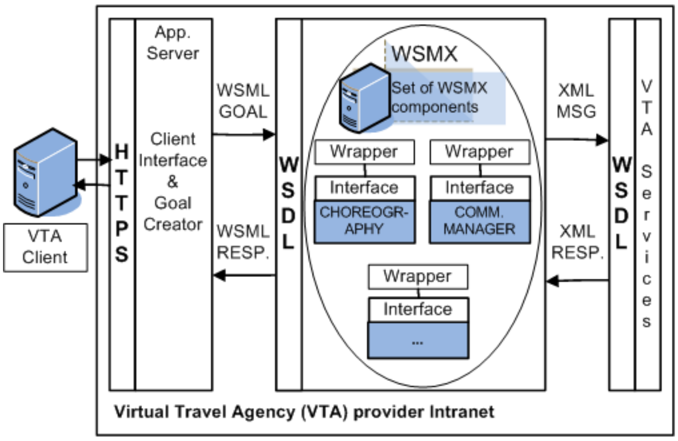


Figure 1: Architecture of the prototype

WSMX acts as a transparent, intermediary layer between interacting parties. All semantic descriptions are provided on top of existing provider's syntactic services, making providers unaware of this semantic layer. No changes are involved in providers' services and native data formats are preserved. User's desires are expressed via web forms that are mapped to appropriate Goal separating him from WSML representation.

Once these semantic descriptions are in place, the merits of Semantic Web services and WSMX can be leveraged. WSMO Choreography allows the declaration of a complex requirement on the actions that client is willing to take or provider is willing to provide. For instance, the client might express through the web forms interface his goal to book a flight from Dublin to Munich on a given data below certain price, then to arrange a shuttle or train from the airport and finally to make a reservation in the hotel situated within 5 km range from the city center. This goal is mapped to the WSMO Goal template, that is populated with input instances like origin and destination airport, final destination and requirements regarding the hotel. Then this Goal is submitted to WSMX, where provider matching this Goal is discovered and according to Goal and Web service choreography communication is carried out. The choreography specifies the execution path of the given partner, which boils down to message exchange patterns. To ensure that the given message exchange is legal, logic formulas are utilized as the transition guards before the given message can be dispatched or received.

Serious advantages of our platform can be also identified in the area of mediation both on the data and process level. The client and discovered Web service might use a different conceptualization in their ontologies which leads to ambiguities that may hamper if not make unfeasible their communication. Data and process Mediation allows these mismatches to be overcome and enable partner communication despite of their data and behaviour differences.

The Data Mediator executes bidirectional mappings between the ontologies, using mapping rules previously defined. The Process Mediator [1] tackles mismatches in partners' choreographies employing logic reasoning in order to evaluate transition rules and determinate if the mismatches can be mitigated. Whilst from the client point of view all required data is sent in a single message, on the VTA Web services side it is the contrary, i.e. there are specialized entrypoints, to which, parts of the client's messages has to be delivered.

3. CONCLUSIONS

We believe that developed prototype for VTA case study is a viable, efficient and dynamic approach. The system allows the expression of goals by the client using web forms that in turn are mapped to WSMO Goals which allows them to be executed by WSMX. User does not have to visit multiple web sites, but can use one portal that aggregates multiple tourist services and can be extended with new ones.

4. ACKNOWLEDGEMENTS

This material is based upon works supported by the Science Foundation Ireland under Grant No. SFI/02/CE1/I131 and EU funding under the DIP project (FP6 - 507483).

5. REFERENCES

[1] E. Cimpian and A. Mocan. WSMX Process Mediation Based on Choreographies. In *Proceedings of the First International Workshop on Web Service Choreography and Orchestration for Business Process Management*, September 2005.

[2] E. Motto, J. Domingue, L. Cabral, and M. Gaspari. IRS-II: A Framework and Infrastructure for Semantic Web Services. In *Proceedings of the second International Semantic Web Conference Sanibal Island, FL, USA*, pages 306–318, 2003.

[3] J. Scicluna, A. Polleres, and D. Roman. Ontology-based Choreography and Orchestration of WSMO Services. WSMO Working Draft, 2006.

Ontology Mapping for Learning Objects Repositories Interoperability

Amel Bouzeghoub
GET/INT, INF Dpt.
9, rue Charles Fourier 91011
Cedex, Evry, France
+33(0)1.60.76.47.14,
Amel.Bouzeghoub@int-evry.fr

Abdeltif Elbyed
GET/INT, INF Dpt.
9, rue Charles Fourier 91011
Cedex, Evry, France
+33(0)1.60.76.42.90,
Abdeltif.Elbyed@int-evry.fr

ABSTRACT

In order to deal with the need of sharing learning objects within and across learning object repositories most of the recent work argue for the use of ontologies as a means for providing a shared understanding of common domains. But with the proliferation of many different ontologies even for the same domain, it become necessary to provide mapping process to perform interoperability. Two key issues must be addressed: the first one is to provide help to users describing and searching resources by organizing the knowledge covered by the learning resources and the second one is to define educational systems interoperability mechanisms to create a virtual learning space. Although many efforts in ontology mapping have already been carried out, few of them use resources properties to generate relations between local concepts and discover mapping dynamically.

Categories and Subject Descriptors

Software/Software_Engineering/Interoperability.

General Terms

Algorithms.

Keywords

Ontology Mapping, Semantic Interoperability, Multi-Agent Systems, Semantic Web, Learning Resources, Web-Based Educational Systems

1. INTRODUCTION

Ontologies offer a great potential in higher education providing in particular the sharing and reusing of information across educational systems and enabling intelligent and personalized learner support. The increased functionality that ontologies imply will bring new opportunities to e-learning. Learners will be able to interact with distant educational systems easily and in a personalized way. An overview of ontologies for education field and an initial report on the development of an ontology-driven web portal O4E are presented in [2].

We propose in this paper an algorithm which is applied on an existing Web Based Educational Systems (WBES) - developed in our team [1] - that allows learners and teachers searching, adding and composing new resources in a local repository. To facilitate resources exchange with other WBES it becomes necessary to find solutions allowing the cooperation between various repositories of learning resources. The user may seek resources out of his/her private reference ontology. The problem is that the comprehension of a new classification (a new ontology) is expensive and does not constitute a justified investment. It is thus necessary to propose mechanisms to permit the user to access to resources of other repositories in a transparent way using his/her favorite WBES (and the associated shared reference ontology).

The particularity of the algorithm is that (i) it focuses on dynamic ontology mapping using a multi-agent system, (ii) it uses information on the resources to enrich the local ontology by generating semantic relations between local concepts (iii) it is based on inference rules to compare the ontologies' concepts. These inference rules may be general ones (i.e. domain independent) or more specific rules (i.e. domain dependent) added by an expert. This flexibility allowed the algorithm to be applied to other domains.

In this paper we introduce a dynamic mapping approach for bridging gaps between learning object repositories based on ontologies. Dynamic ontology mapping means that during a user interaction (query), the mapping system receives a sequence of external concepts and returns the most specific mapping for each concept.

2. ALGORITHM PRINCIPAL

The objective of our approach is to map ontologies dynamically, and only when needed. The system tries to find semantic relations between the user query concepts and the concepts in the target ontologies.

The algorithm to combines different similarity measures to find mapping candidates between two ontologies. We distinguish three main categories of similarity: linguistic similarity, structural similarity and rule-based similarity. Using these different similarities may increase the precision of the results.

In this section we describe the global architecture and the agents' behavior of the mapping process.

2.1 Architecture

The ontology interoperability needs to define mapping between ontologies. In our architecture (figure 1) the mapping process is split up into five levels: (i) resources level, (ii) ontology level, (iii) interface level where we can find, the user and the ontology agents (OA) which generate new ontologies enriched with additional relations, (iv) simulation level and (v) domain expert level.

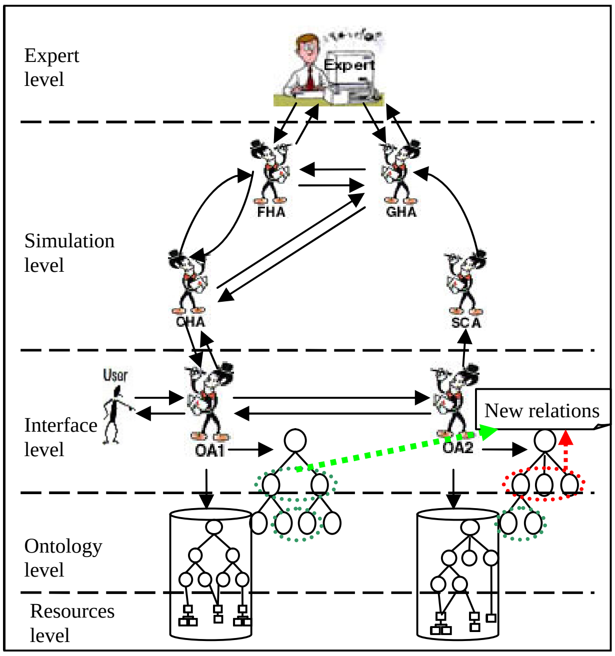


Figure 1. Architecture approach for mapping process.

2.2 Mapping Process

The algorithm begins by generating information from the ontology. The ontology agent OA uses the instances (resources) comparisons for deducing semantic relations between concepts (convergence, divergence) of the same ontology. The OA agent which intercepts a user query generates all possible relations between the query concepts and sends both concepts and these relations to all other OA agents.

The simulation level contains four agents: SCA (Similarity Computation Agent), GHA (Generation Hypotheses Agent), FHA (Filtration Hypotheses Agent) and CHA (Choice Hypotheses Agent). We describe in the following the general behavior of each simulation agent.

The SCA agent determines similarity values of candidate mappings via different matchers. The first iteration consists in providing a basic similarity between concepts. In this iteration we use linguistic tools [6, 7] to compare concepts' names. In the i^{th} iteration we use the similarity produced in $(i-1)^{th}$ iteration and we apply the inference rules. These inference rules are either rules inferred from structural similarity (deduction rules) or rules proposed by the domain expert (comparison rules).

The GHA agent receives all similarities sent by SCA and it generates hypotheses using inference rules. These hypotheses consist of new correspondences between concepts. The generation of an hypothesis at iteration (i) is based on either the mapping set or the similarities generated previously. Indeed, depending on the similarity value, we generate mapping hypotheses between the couple of concepts which have a similarity value enough important.

The FHA agent studies and filters all hypotheses generated by GHA. The hypotheses which do not verify certain constraints (e.g. structural constraints) are removed. The subset of filtered hypotheses is sent to CHA agent.

The CHA agent choices hypotheses which have the best similarity, using both existing mapping and user feedback.

The final mapping is sent to ontology agents. After several interactions, each OA acquires more knowledge about other OA(s) and defines a set of most relevant OA(s) (i.e. the OA(s) that answer to its needs). This set is called “agent’s accountancies”.

3. CONCLUSION

Various works [3, 4, 5] have been developed for supporting the mapping of ontologies. Most of them are based on syntactic and semantic matching heuristics given by an expert to generate static mapping. None uses deduction rules which can be generated for different application domains. In our mapping approach, we try to use as much as possible available information contained in the ontology to determine dynamically and if necessary the relationship between concepts. This information consists of identifiers names of concept, ontology structure, resources and manual/automatic rules. Resources properties generate new semantic relations between concepts (concepts of the same ontology). In future work, we plan to add other match and techniques in order to resolve more complex mapping problems.

4. REFERENCES

[1] Bouzeghoub A., Defude B., Ammour S., Duitama J.F., Lecocq C.: A RDF Description Model for Manipulating Learning Objects, Proc. IEEE International Conference on Advanced Learning Technologies, Joensuu, Finland, August 2004

[2] Dicheva1, D., Sosnovsky, S., Gavrilova, T., Brusilovsky, P.: Ontological Web Portal for Educational Ontologies. International Workshop on Applications of Semantic Web technologies for E-Learning (SW-EL’05) july 2005

[3] Euzenat J., Le Bach, T., Barrasa J., Bouquet P., De Bo J., Dieng-Kuntz R., Ehrig M., Hauswirth M., Jarrar M., Lara R., Maynard D., Napoli A., Stamou G., Stuckenschmidt H., Shvaiko P., Tessaris S., Van Acker S., Zaihrayeu I.: State of the art on ontology alignment. Deliverable 2.2.3, IST Knowledge Web NoE, 21p., june 2004

[4] Ehrig, M., Sure Y.: Ontology Mapping - An Integrated Approach April 21,2004, ESWS 2004: 76-91

[5] Kalfoglou Y., Schorlemmer M.,Ontology mapping: the state of the art, The Knowledge Engineering Review 18(1):1--31, January 2003

[6] Silva, N., Rocha, J.A.: Ontology mapping for interoperability in semantic web. In Proceedings of the IADIS International Conference WWW/Internet 2003 (ICWI’2003), Algarve, Portugal, 2003.

[7] http://www.wordnet.com

On the relation between trust on input and reliability of output

Mauro Mazzieri
DEIT
Università Politecnica delle Marche
Ancona, Italy
mauro.mazzieri@gmail.com

Aldo Franco Dragoni
DEIT
Università Politecnica delle Marche
Ancona, Italy
a.f.dragoni@univpm.it

ABSTRACT

We are developing a framework to acquire knowledge from a peer-to-peer network and to evolve it maintaining local coherence and recoverability of previously acquired data.

In this context, we give trust to informations and information sources in a novel way. Informations acquired from the web is given a truth value, based on previous valuations. Each peer deal with them using a fuzzy extension of RDF semantics, apply extra entailment rules derived from an ontology, and deal with inconsistencies using belief-revision techniques. The result is a set of consistent and logically closed data that can be visualized and annotated by human users.

The knowledge shared with other peers has trust metadata on them. Trust values comes both from belief revision process outputs and from user annotations.

Keywords

belief revision, Semantic Web, trust

1. INTRODUCTION

Our reference scenario is that of RDF/S data acquired from a P2P network with a grow-only model: after quiescence, every peer has all the available informations about a shared and agreed topic of interest. An example is the RDFGrowth algorithm[8], where topics of interests are defined by an operator called GUED (Group URI Exposing Definition) and each community agree on a common ontology.

The grow-only model is somehow required by the monotonic nature of the semantic web. Problems arise when the data, according to the ontology, became inconsistent. In this case we locally use a belief-revision technique to restore consistency (section 3). The belief revision use a fuzzy extension of RDF semantics to represent trust values as "truth" values. The assignement of those trust values is explained in section 2.

Finally, section 4 summarize how new trust metadata are attached to information shared with other peers.

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

2. ACQUIRED KNOWLEDGE

2.1 Trust values

We make a distinction between the **source** of a statement, i.e. the peer who produced the data, and the **peer** from which we actually downloaded the data.

The identity of a peer can be established with a digital signature, either on the document containing data (i.e. an RDF graph) or over singles data clusters (MSGs[9]): the former methos is used by peers, the latter by sources.

The attribution of a trust value to incoming data obeys the following criteria.

Explicit attribution An application-specific property can be used by a source to attach trust values to reified RDF statements, named graphs[2] or MSGs[9]. The actual value is internally weighted with the trust on the source inferred from previous experiences.

Source-based default The trust on statements with no explicit value is equal to the trust on their source.

Peer-based default If the source of a statement cannot be established, it is given a value equal to the trust of the peer from which we downloaded it.

2.2 Knowledge representation

Internally we represent data using a fuzzy semantic extension of RDF and RDF Schema[6, 7].

At the syntactic level, this extension adds a value (i.e. a number) to a triple ⟨subject, predicate, object⟩. The added element has a syntactic nature different from the others: it is not an element of the domain of the discourse, but a property related to the formalism used by the language to represent uncertainty and vagueness.

At the semantic level, the *extension* of a predicate (defined in [5] as a set of couples ⟨subject, object⟩) becomes a fuzzy set.

Analogously, *fuzzy RDF Schema* is based on the definition of a fuzzy *class extension* as a fuzzy subset of domain's elements. Not without some troubles, also domain, ranges, subproperties and subclasses are defined, allowing the representation of simple fuzzy ontologies.

3. THE BRIDGE BETWEEN MONOTONIC AND NONMONOTONIC WORLDS

The semantic web model has a strict monotonic discipline[5] and no global coherence requirement. Howevere, when we give RDF semantic meaning according to an ontology, we can have inconsistencies: for example, two different sources can give a different value to a functional property of a resource.

Our model is to deal with possible inconsistencies maintaining an external grow-only behavior, while internally using belief revision techniques to maintain local coherence.

We make use of a belief revision technique that drops the AGM[1] principle of priority to incoming information, as data come asynchronously from peers. Instead, we use a consistency-based approach[3] to extract a subset of **KB** that is maximal in term of fuzzy cardinality and minimal in term of inconsistence. This maximal subset may not contain the incoming data.

As there is a tradeoff between those requirements, a single parameter is maximazed: the *weighted cardinality value* fuzzy cardinality $*$ (1 $-$ fuzzy inconsistence).

The entire framework uses 3 correlated knowledge bases:

KB, a monotonic (and possibly inconsistent) knowledge base that grows with informations taken from the peer-to-peer network.

B, a maximal consistent subset of **KB**;

Th(B), the deductive closure of **B**, representing the current belief set of the agent, actually visualized by the user interface.

4. OUTPUT RELIABILITY VALUES

4.1 User annotations

The user can interact with the data in **Th(B)**, allowing a finer-grade control on which data is relevant and deserves to be visualized. In a way similar to that of [4], each statement can be either visualized, hidden because it is considered *unreliable*, or hidden because it is considered *irrelevant.* The inferred trust rating on statements is graphically visualized and can be overridden.

As explained in the following, each user interaction causes variations on **KB** statements truth values, and can eventually change the composition of **B** and **Th(B)**.

Moreover, the user can set configuration options to state a self-judgement about his knowledge of the topic and his ability to make judgements.

4.2 Shared outputs

The composition of **Th(B)** and the user annotation on its elements allows to give a feedback to **KB**, updating information metadata before they are shared.

If the user has made explicit his trust on a statement, this statement has in **KB** a truth value equal to user trust, and when it is shared it has the user's digital signature.

Statements with a source signature are transmitted as-is, thus preserving original truth value (unless the user overrided the value). A new value foa a source's reliability is calculated as the ratio between the fuzzy cardinality of statements in **Th(B)** and the cardinality of the set of statements from that source. However, this value is used only internally as a weight for trust value from that source.

The same happens for peers: statements are shared without an explicit trust value; an updated peer reliability value is calculated as the ratio between the fuzzy cardinality of statements in **Th(B)** and the cardinality of set of statement from the peer.

5. REFERENCES

[1] C. E. Alchourrón, P. Gärdenfors, and D. Mankinson. On the logic of theory change: partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50:510–530, 1985.

[2] J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named graphs, provenance and trust. Technical report, HP Labs, 2004.

[3] A. F. Dragoni. Belief revision: From theory to practice. *The Knowledge Engineering Review*, 12(2), 1997.

[4] Y. Gil and V. Ratnakar. Trusting information sources one citizen at a time. In *Proc. of International Semantic Web Conference (ISWC 2002)*, pages 162–176, 2002.

[5] P. Hayes. RDF Semantics. W3C recommendation, World Wide Web Consortium, 10 February 2004.

[6] M. Mazzieri. A fuzzy RDF semantics to represent trust metadata. In *1st Workshop on Semantic Web Applications and Perspectives (SWAP2004)*, Ancona, Italy, 10th December 2004.

[7] M. Mazzieri and A. F. Dragoni. A fuzzy semantics for semantic web languages. In *ISWC Workshop on Uncertainty Reasoning for the Semantic Web*, volume 173 of *CEUR Workshop Proceedings*, pages 12–22, Galway, Ireland, November 7 2005.

[8] G. Tummarello, C. Morbidoni, J. Petersson, F. Piazza, and P. Puliti. RDFGrowth, a P2P annotation exchange algorithm for scalable semantic web applications. In *Mobiquitous*, Boston, MA, August 23–25 2004.

[9] G. Tummarello, C. Morbidoni, P. Puliti, and F. Piazza. Signing individual fragments of an RDF graph. In *Proc. of the World Wide Web Conference (WWW2005)*, 2005.

TMwiki – a Collaborative Environment for Topic Map Development

Bernd Markscheffel
Technische Universität Ilmenau
PF 10 05 65
98684 Ilmenau
+49 (0)3677 / 69 4053
Bernd.Markscheffel@TU-Ilmenau.de

Hendrik Thomas
Technische Universität Ilmenau
PF 10 05 65
98684 Ilmenau
+49 (0)3677 / 69 4053
Hendrik.Thomas@TU-Ilmenau.de

Dirk Stelzer
Technische Universität Ilmenau
PF 10 05 65
98684 Ilmenau
+49 (0)3677 / 69 4040
Dirk.Stelzer@TU-Ilmenau.de

ABSTRACT
TMwiki is a prototype for a collaborative environment that helps to reduce the complexity of developing large semantic structures. It features standard Wiki-functionality and gives users access to several useful Topic Map tools. It comprises a powerful browser component which provides a topic search as well as a collaborative editing section coupled with a revision history for Topic Maps.

Categories and Subject Descriptors
D.4.9 [**Operating Systems**]: Systems Programs and Utilities – *Php*. H.3.7 [**Digital Libraries**]: collection. I.2.4 [**Artificial Intelligence]:** Knowledge Representation Formalisms

General Terms
Algorithms

Keywords
Topic Maps, Digital Libraries, Wiki, Collaboration

1. INTRODUCTION
The major objective of digital libraries is to provide users with an effective access to information resources that are represented in various formats. Issues in the implementation of Digital Libraries have been widely addressed in the literature. A key challenge is to provide effective information retrieval mechanisms for heterogeneous information resources with different levels of complexity. Semantic technologies, e.g. Topic Maps, frames or semantic mark-up languages, can be used to structure heterogeneous information resources. Thus, the complex relationships of information resources can be modelled with the help of semantic technologies. This helps to improve information retrieval in digital libraries. We have developed TMwiki as part of the DMGLib (Digital Mechanism and Gear Library) project [1]. The objective of DMGLib is to develop a digital library for mechanical engineering. DMGLib contains a vast amount of digital documents representing more than 1000 gear models, 100 machines, 3500 photographs, 100 videos and animations, 400 books published before 1898 and nearly 10000 mechanical engineering documents, such as technical reports, patent specifications, research papers and books [1]. These digital documents are represented in various different formats and media. DMGLib is developed by approx. 30 engineers at 8 different sites.

A universal solution to the problem of structuring large information spaces demands for an interdisciplinary, distributed, and collaborative approach [2, 3]. TMwiki is a first step towards building such a collaborative environment. It also is an essential element of a sophisticated navigation tool in the DMGLib-Portal.
We implement a semantic meta-layer that helps to integrate different terminologies, languages, research paradigms and historic eras in which the digital documents included in the library have been developed. This facilitates information retrieval in voluminous and complex libraries.

2. TMwiki
TMwiki (www.topic-maps.org) is a prototype for a collaborative environment that supports the development of Topic Maps. It is a web application which allows users to add and edit content. TMwiki is based on DokuWiki [4], a standards compliant and simple to use Wiki engine.
TMwiki supports collaborative work by providing numerous features. Besides standard Wiki-features such as collections of relevant resources (Topic Map software, Topic Map samples, glossary, web pages, blogs), mailing lists and discussion areas, RSS feeds and a tutorial section, it also provides access to some dedicated Topic Map tools, e.g. MERLINO [5], a tool for semi automated generation of occurrences in topic maps and a TopicMapVisualizer (see below). TMwiki also offers a sandbox for exploring and testing these applications.

2.1 TopicMapVisualizer - TMV
The collaborative use of Topic Maps requires a flexible and powerful navigation engine. TMV is a generic Topic Map browser which is used in TMwiki as an interface for displaying

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 11 – 14 June 2006

and navigating in Topic Maps. TMV consists of the following components:

Overview

The overview section helps users to get a general idea of the Topic Map that is currently displayed. The overview section displays all major structure elements (topics types, association types, scope topics and role specifications used in associations) of the current Topic Map.

Topic Map Graph Viewer

The Topic Map Graph Viewer enables user-friendly navigation and browsing in the current Topic Map. When a specific topic has been found by a search query, this topic is displayed as centroid of the relevant Topic Map. Associated topics are grouped around it. They are displayed as circles. When association types are detected they are displayed as labels at the connection line between the topics. The MouseOver-function displays an information box showing topic id, baseNames, class name and references to relevant sources. The standard viewing option can be extended to a broader view that shows all associations.

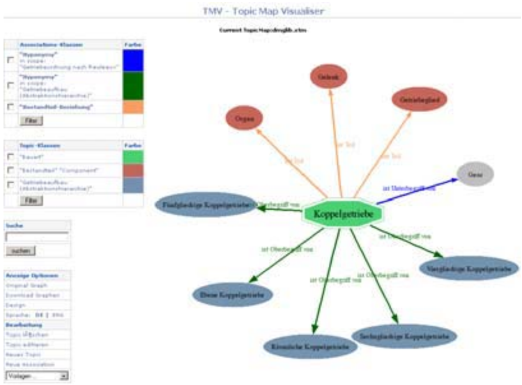


Figure 1: TMV screen with Class Filter section

Navigation in the current Topic Map is very easy. When users click on an associated topic, the Topic Map Graph Viewer switches to a new graph structure that displays the current topic as centroid.

Class Filter

The Class Filter helps users to select specific classes in complex Topic Maps. Users may select specific views of the semantic structure by hiding classes that are currently not required.

Topic Map Search

The search option allows users to perform a case sensitive, truncated search in Topic IDs, BaseNames, and Occurrences in a Topic Map.

Information Table

The Information Table displays all available attributes of the current topic: topic names and their scopes, subject identity of the topic, classes of the topic, occurrences (references and internal data) and their scopes and classes, associated topics including role specifications of all members and association classes, all topics that are instances of the current topic, all topics that reference the current topic as a scope in one of their BaseNames, and all topics that reference the current topic in one of the occurrences or as scope in an involved association.

2.2 Editor

The editor of TMwiki helps users to upload and edit any XTM compliant Topic Map. Several short-cut-buttons (see Figure 2) support the design of a well-formed Topic Map. TMwiki also checks the edited code and validates it on the basis of the XTM version=1.0 DTD. If a topic map has an error, a message is displayed in the upper field of the editing screen. The revision history allows users to track recent changes.

Figure 2: Editing section of TMwiki

3. CONCLUSION

The major advantage of TMwiki is its ability to support the collaborative development of Topic Maps. TMwiki offers a powerful navigation interface coupled with a user-friendly editing section. It helps users to create and maintain semantic structures.

However, TMwiki is a prototype. It has to be tested more thoroughly. This will be done during the next months in the DMGLib project.

4. REFERENCES

[1] Torsten Brix, Ulf Döring, Sabine Trott: *DMGLib - ein moderner Wissensraum für die Getriebetechnik*. In: Roswitha Moes, Cornelia Plott (Eds.): *Knowledge eXtended. Die Kooperation von Wissenschaftlern, Bibliothekaren und IT-Spezialisten.* Jülich 2005, 251-262.

[2] Arttu Valo, Eero Hyvönen, Ville Komulainen: A Tool for Collaborative Ontology Development for the Semantic Web http://www.seco.tkk.fi/publications/2005/komulainen-valo-et-al-a-tool-for-collaborative-2005.pdf

[3] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke. OntoEdit: Collaborative Ontology Engineering for the Semantic Web. In: Proceedings of the first International Semantic Web Conference 2002 (ISWC 2002), June 9-12 2002, Sardinia, Italia, Springer, LNCS 2342, 221-235.

[4] Andreas Gohr: *Wiki:dokuwiki*. http://wiki.splitbrain.org/wiki:dokuwiki

[5] Bernd Markscheffel, Hendrik Thomas, Dirk Stelzer: *Merlino - a prototype for semi automated generation of occurrences in topic maps using internet search engines.* In: *Posters and Demos*, 2nd European Semantic Web Conference 2005, Heraklion, Greece, 29.05.05 - 01.06.05, 25 -27.

Web Service Ranking in Service Networks

John Gekas
University of Essex
Wivenhoe Park, Colchester
Essex, UK.
0044 1206 873805
jgekas@essex.ac.uk

ABSTRACT

In this paper, we present the concept of *web service ranking*: a service rank is a quantitative metric that in some way shows the “importance” of a service within a web service network. The ranks we briefly introduce here are based on a variety of metrics, borrowed from graph network and social network analysis, and thus the “importance” of a web service is defined differently in the context of each ranking method. We also attempt to explain how web service ranking can be used in the context of web service discovery and composition, so that successful solutions can be found with traversing as little of the web service network as possible.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures – *Domain-specific architectures.* G.2.2 [**Discrete Mathematics**]: Graph Theory – *Graph algorithms, Network problems*.

General Terms

Algorithms, Measurement.

Keywords

Web service networks, web service ranking, web service discovery-composition, graph networks, semantic web.

1. INTRODUCTION – RELATED WORK

In this paper we present the idea of applying network analysis mechanisms in networks of semantic web services: we believe that information derived from service network link analysis can prove highly useful in order to provide effective service discovery and composition mechanisms. In this context we present the notion of *web service ranking*: ranking measures given to web services belonging to a particular service network, measuring the “importance” of the service within the network. A number of ranking mechanisms can be employed, depending on the analysis criteria used in order to calculate the ranks. The ranking systems we briefly present in this paper are based on social network and graph network analysis methods, and are mainly aimed towards connectivity measures and link analysis within the web service network. We assume that web service ranking takes place within a directory of web services, for which compatible, semantic descriptions are available; available semantic information about each web service can allow us to analyse whether different services are “compatible” and can be “linked” – an important concept for performing network link analysis.

The motivation/vision behind this work is that such ranking systems could be used by service discovery and composition systems that operate as service directory search mechanisms, attempting to “extract” successful solutions to a request by traversing as little of the overall search network as possible. Thus, the ranking systems should serve as a “heuristic” guide towards successful solutions. In this context, we believe that the use of social network analysis methodologies can provide invaluable help in facilitating resource-efficient querying mechanisms for service discovery and composition. Experimental evaluation of which particular ranking methods are the most effective towards the highest service discovery performance is out of the scope of this paper. However, a detailed performance evaluation of some of the proposed ranking systems can be found at previous work done by the authors [2], [3].

Even though a number of research approaches have addressed the problems of web service discovery and composition, the areas of web service network analysis and web service ranking have not been directly approached. However, [4] propose a service composition approach based on a best-first graph search algorithm, where the services that are evaluated first by the algorithm are the ones that “can lead to the largest number of data types” – which vaguely encapsulates the notion of a service rank as a connectivity measure. Furthermore, [1] propose an architecture supporting similar service composition approaches, where the best-first composer can be led by “service ranking mechanisms specified by the user”. The composer makes use of a “priority queue” (the *heap*), where services are added in descending order, depending on their rank. The particular approach mainly focuses on the architectural design of such a service directory and the specification of a query language in which the ranking systems can be specified, rather than the service ranking mechanisms themselves. Finally [5] is a search engine project specialising in searching within directories of java applications/classes. The search algorithm is based on the *rank* of a java component, where the rank is calculated as the part of the directory the component can be linked to.

We believe that network analysis can prove extremely useful when applied in the context of semantic web service networks: using the appropriate analysis tools, web services can be ranked in terms of “importance” or “usefulness” within a directory, so that the discovery of successful solutions can be performed in a resource-effective way. Finally, to our knowledge, no other analytical approach has applied concepts borrowed from network analysis to web service directories.

2. WEB SERVICE RANKING

The ranking metrics we describe can be categorised in two different ways: (a) *local* and *global*, depending on whether local or global network knowledge is needed for their estimation, and (b) *absolute* and *relative*, depending on whether the measurement

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 11th - 14th June, 2006

is of absolute scope or refers to a particular client request. Based on this categorisation, we can see that there are four different combinations a service rank can fall under. Our service metrics are based on graph network analysis metrics that have been used is other research areas, such as social network analysis and bibliometrics. Furthermore, we have to note that these ranks rely on the analysis of the link structure of the service network (there is only one exception – *data type semantic similarity*). In the context of a web service network, a *forward link* is defined as when the output data provided by a particular service is sufficient in order to call another service. The following table shows the network metrics used, along with the category combination they fall into, based on the above categorisation:

Table 1: Categorisation of Web Service Ranks.

	Local	**Global**
Absolute	Absolute Degree(ADR), Hubs-Authorities(HAR), QoS Rank(QR).	PageRank(PR), Closeness(CR), Betweenness(BR).
Relative	Relative Degree(RDR), HITS, data type semantic similarity(SSR).	Depth-Limited Walks(WR), Flow(FR).

We will not go into a detailed description of each of the above ranks, due to space restrictions. However, it would be useful to briefly describe some of them and some of the cases they can be useful in:

Degree-based Ranks: The ADR, RDR and PR ranks are based on the degree of a web service - the number of services a web service can "feed", as a normalised percentage. RDR shows the part of the ADR that belongs in the semantic data type category specified in the request. ADR and RDR are both simple and "light" estimates of how important a service is, since they can be calculated directly, without requiring global knowledge of the service network. PR (similar to the PageRank algorithm used by the Google search engine) of course is global by nature, making it "heavier" to estimate, but also richer in informational value.

Hubs-Authorities – based Ranks: HAR and HITS both examine the relation between the number of services that link to a specific service (in-degree) and the number of services that service links to (out-degree). This is important, since there are web services and semantic data types whose one degree type tends to be much higher than the other – e.g. web services that operate on *non-functional parameters* (like e.g. *Transaction Confirmation ID*). Such high degrees would be able to attract the composer even though they could potentially lead to dead-ends in the service network, and thus should be identified.

Non-Functional Ranks: A number of the presented service ranks focus on the non-functional aspects of service composition. For instance, QR (which is calculated with regard to a specific QoS attribute) has the form of a percentage ranging from 0 to 1: this rank examines the specified QoS value of the services a service links to, estimates an average, and places it on a normalized [0,…,1] scale compared to the range of the QoS values found within the service network. Such a rank can be extremely useful when a service composition request explicitly declares QoS restrictions. Also, the FR and WR ranks examine how many alternative routes exist between two web services, which can be useful when we are interested in the reliability of a workflow solution (i.e. if a part of the solution is unavailable, will the solution fail?).

Non-Connectivity Ranks: Even though most ranks presented here are related on connectivity aspects of the web service networks, this does not always have to be the case. A useful rank that is not related to connectivity degrees is the SSR (semantic similarity rank): this rank evaluates how semantically related the data provided by two web services are. For this purpose, we assume semantic data items are defined in some form of classification/ontology – SSR is estimated as the graph distance in the ontology graph, between the data types in question.

3. DISCUSSION

The above list of web service ranks is by no means complete: similar ranks can be defined with regards to any property/attribute inherent to web service networks, depending of course on the way the rank is intended to be used. In our work, the particular ranks were chosen because they give an idea of the connectivity structure of the network.

In our work, the above ranks are used for web service discovery and composition: the service composition mechanism is defined as a graph search algorithm, that traverses the service network (search space) with the purpose of extracting successful solutions by searching as little of the search space as possible. In this sense, the composer makes use of a *Priority Queue*, where web services are added in series of how "important" and "useful" they are considered to be: this assessment is made based on the rank used at each particular case.

A detailed presentation of which ranks seem to perform better in the context of service composition is out of the scope of this paper – however, we can claim that web service ranks that measure attributes relatively to the request seem to perform higher than the absolute ones. A detailed experimental setting and performance analysis can be found at [2] and [3].

4. REFERENCES

[1] Walter Binder, Ion Constantinescu, Boi Faltings: Directory Support for Large-Scale, Automated Service Composition. Software Composition 2005: 57-66.

[2] John Gekas, Maria Fasli: Automatic Web Service Composition Based on Graph Network Analysis Metrics. OTM Conferences (2) 2005: 1571-1587.

[3] John Gekas, Maria Fasli: Service Network Structure Analysis for Web Service Discovery and Composition. In proc. BASEWEB 2006.

[4] Seog-Chan Oh, Byung-Won On, Eric J. Larson, Dongwon Lee: BF*: Web Services Discovery and Composition as Graph Search Problem. EEE 2005: 784-786.

[5] Silvestri F., Puppin D., Laforenza D., Orlando S.: Toward a Search Engine for Software Components. IEEE Web Intelligence, Beijing, China, September 20-24, 2004.

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In WWW Conference, volume 7, 1998.

Semantic Wikipedia

Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller, Rudi Studer
Institute AIFB, University of Karlsruhe (TH)
76128 Karlsruhe, Germany
{voelkel,kroetzsch,vrandecic,haller,studer}@aifb.uni-karlsruhe.de

ABSTRACT

Wikipedia is the world's largest collaboratively edited source of encyclopaedic knowledge. But its contents are barely machine-interpretable. Structural knowledge, e. g. about how concepts are interrelated, can neither be formally stated nor automatically processed. Also the wealth of numerical data is only available as plain text and thus can not be processed by its actual meaning.

We provide an extension to be integrated in Wikipedia, that allows the typing of links between articles and the specification of typed data inside the articles in an easy-to-use manner.

Enabling even casual users to participate in the creation of an open semantic knowledge base, Wikipedia has the chance to become a resource of semantic statements, hitherto unknown regarding size, scope, openness, and internationalisation. These semantic enhancements bring to Wikipedia benefits of today's semantic technologies: more specific ways of searching and browsing. Also, the RDF export, that gives direct access to the formalised knowledge, opens Wikipedia up to a wide range of external applications, that will be able to use it as a background knowledge base.

Keywords

Semantic Web, Wikipedia, RDF, Wiki

1. INTRODUCTION

This paper describes an extension to be integrated in Wikipedia, that enhances it with Semantic Web [1] technologies. Wikipedia, the free encyclopaedia, is well-established as the world's largest online collection of encyclopaedic knowledge, also being an example of global collaboration within an open community of volunteers.

Using Wikipedia currently means *reading* articles—There is no way to automatically gather information scattered across multiple articles, like "Give me a table of all movies from the 1960s with Italian directors". Although the data is quite structured (each movie on its own article, links to actors and directors), its meaning is unclear to the computer, because it is not represented in a machine-processable, i. e. formalised way.

To let the huge and highly motivated community of Wikipedians render the shared factual knowledge of Wikipedia machine-processable, we face several challenges: In addition to technical aspects of this endeavour, the main challenge is to introduce semantic technologies into the established usage patterns of Wikipedia. We propose small extensions to the wiki link syntax and an enhanced article view to show the interpreted semantic data to the user. Powerful inline queries turn parts of a page into a dynamically updated list or table. These queries have the potential to replace the many hand-crafted lists (e. g. cities in Europe).

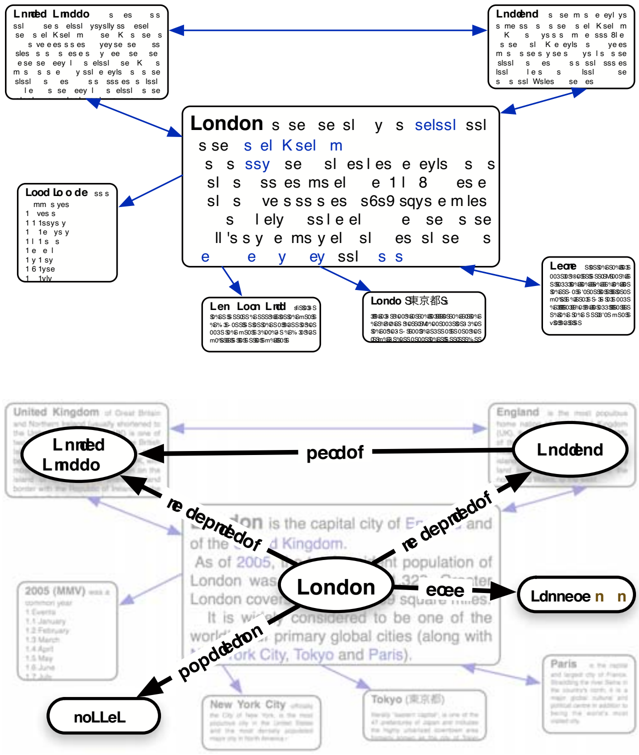


Figure 1: Currently there are pages and links (above), we feature concepts and data connected by relations (below).

We expose Wikipedia's fine-grained human edited information in a machine-readable way by using the W3C standards on RDF, XSD, RDFS, and OWL. This opens new ways to improve Wikipedia's capabilities for querying, aggregating, or exporting knowledge, based on well-established Semantic Web technologies. We hope that Semantic Wikipedia can help to demonstrate the promised value of semantic technologies to the general public.

The primary goal of this project is to supply an implemented ex-

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

tension to be actually introduced into Wikipedia in the near future. The implementation is rapidly developing, and the software can be tested online at `http://wiki.ontoworld.org`.

2. IDEA

Our primary goal is to provide an extension to MediaWiki which allows to make important parts of Wikipedia's knowledge machine-processable with as little effort as possible[3]. Since our system is conceived as an extension of MediaWiki it adheres to these core wiki principles—often referred to as the "wiki way" [2]—with all the advantages and disadvantages that this brings.

We designed the following key elements for our annotations:

- *categories*, which classify articles according to their content,
- *typed links*, which classify links between articles according to their meaning, and
- *attributes*, which specify simple properties related to the content of an article.

Categories already exist in Wikipedia, though they are mainly used to assist browsing. Typed links and attributes are novel features that are explained below and detailed in subsequent sections.

We restricted the annotations to have as their subject always the topic of the current page. Thus it is not possible to make statements about a topic elsewhere then on the topic's page. This helps e. g. to locate erroneous statements.

2.1 Relating Concepts with Typed Links

Typed links are obtained from normal links by slightly extending the way of creating a hyperlink between articles, as illustrated in Figure 1. As for the Web in general, links are arguably the most basic and also most relevant markup within a wiki, and their syntactic representation is ubiquitous in the source of any Wikipedia article. The introduction of typed links thus is a natural consequence of our goal of exploiting existing structural information. Through a minor, optional syntax extension, we allow wiki users to create (*freely*) *typed links*, which express a *relation* between two pages (or rather between their respective subjects).

In order to explicitly state that London is the capital of England, in the "London" article one just extends the existing link to `[[England]]` by writing `[[is capital of::England]]`. This states that a relation called "is capital of" holds between "London" and "England." Typed links stay true to the wiki-nature of Wikipedia: Every user can add an arbitrary type to a link or change it. Of course existing link types should be used wherever applicable, but a new type can also be created simply by using it in a link. To make improved searching and similar features most efficient, the community will have to settle down to re-use existing link types. As in the case of categories, we allow the creation of descriptive articles on link types to aid this process.

Note how typed links integrate seamlessly into current wiki usage. Semantic MediaWiki places semantic markup directly within the text to ensure that machine-readable data agrees with the human-readable data of the article. The notation we have chosen makes the extended link syntax largely self-explicatory.

In the Semantic Wikipedia, even very simple search algorithms would suffice to provide a precise answer to the question "What is the *capital of England*?" In contrast, the current text-driven search returns only a list of articles for the user to read through. Details on how the additional type information can be added in an unobtrusive and user-friendly way are given in the next section.

2.2 Data Values as Concept Attributes

Attributes provide another interesting source of machine readable data, which incorporates the great number of data values in the encyclopedia. Typically, such values are provided in the form of numbers, dates, coordinates, and the like. For example, one would like to obtain access to the population number of London. It should be clear that it is not desirable to solve this problem by creating a typed link to an article entitled "7421328" because this would create a unbearable amount of mostly useless number-pages whereas the textual title does not even capture the intended numeric meaning faithfully (e.g. the natural lexicographic order of titles does not correspond with the natural order of numbers). Therefore, we introduce an alternative markup for describing attribute values in various datatypes.

In order for such extensions to be used by editors, there must be new features that provide some form of *instant gratification*. Semantically enhanced search functions improve the possibilities of finding information within Wikipedia. Additionally, Wikipedia's machine-readable knowledge is made available for external use by providing an RDF export of each page. This enables the creation of additional tools to leverage Wikipedia contents and re-use it in other contexts. Thus, in addition to the traditional usage of Wikipedia, a new range of services is enabled inside and outside the encyclopaedia. Experience with earlier extensions, such as Wikipedia's category system, assures us that the benefits of said services will lead to a rapid introduction of typed links into Wikipedia.

2.3 Inline Queries

Semantic MediaWiki offers inline queries. In edit mode, the user can specify the query using a wiki-like syntax. In normal view-mode, the results of the query are displayed. The expressivity is less than SPARQL and the current implementation uses MySQL 4.1 queries, as we could not find a scalable, 100% open-soure (i. e. not Java) triple store with SPARQL and inferencing support. As an example, we show a query asking for all actors born in Boston: `<ask>[[Category:Actor]] [[born in::Boston]]</ask>`.

3. CONCLUSIONS AND OUTLOOK

We have demonstrated that the system provides many immediate benefits to Wikipedia's users, such that an extensive knowledge base might be built up very quickly. The emerging pool of machine accessible data presents great opportunities for developers of semantic technologies who seek to evaluate and employ their tools in a practical setting. In this way, Semantic Wikipedia can become a platform for technology transfer that is beneficial both to researchers and a large number of users worldwide, and that really makes semantic technologies part of the daily usage of the World Wide Web.

Acknowledgements. This research was partially supported by the European Commission under contract IST-2003-506826 "SEKT," and FP6-507482 "Knowledge Web Network of Excellence," and by the German BMBF project "SmartWeb."

4. REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, (5), 2001.

[2] W. Cunningham and B. Leuf. *The Wiki Way. Quick Collaboration on the Web*. Addison-Wesley, 2001.

[3] M. Völkel, M. Krötzsch, D. Vrandecic, and H. Haller. Semantic wikipedia. In *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, May 23-26, 2006*, MAY 2006.

CCBR Ontology for Reusable Service Templates[1]

Charlie Abela, Matthew Montebello
Department of Computer Science and AI
University of Malta
+356 2590 7295
{charlie.abela, matthew.montebello}@um.edu.mt

ABSTRACT
We present the motivation and design of CCBROnto, an OWL Ontology for Conversational Case-Base Reasoning (CCBR). We use this ontology to define cases that can eventually be stored, retrieved and reused by a mixed-initiative approach based on CCBR. We apply this technique for retrieving Web Service Composition templates.

Categories and Subject Descriptors
I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

General Terms
Design, Implementation

Keywords
CCBR, Ontologies, OWL, Web Services

1. INTRODUCTION
Web Services composition is usually interpreted as the integration of a number of services into a new workflow or process. A number of compositional techniques have been researched [9,10] that attempt to address service composition by composing web services from scratch while ignoring reuse or adaptation of existing compositions or parts of compositions. Furthermore composing web services by means of *concrete* service interfaces leads to tightly-coupled compositions in which each service involved in the chain is tied to a web service instance. This approach may lead to changes in the underlying workflow which range from slight modifications of bindings to whole redesigning of parts of the workflow description. Therefore we interpret services at an abstract level to facilitate their independent composition. Infact our approach is more similar to [8,11,12], which use pre-stored abstract workflow definitions or templates in their composition framework. Abstract workflows allow for more generalisations and a higher level of reusability [5]. The use of such templates can be thought of as a pre-processing stage towards service discovery and composition, whereby abstractly defined workflow knowledge can be concretely bound to actual services that satisfy a template. To make effective reuse of such templates we have considered CCBR [6]. This extends from CBR and allows for partial definition of the problem by using a mixed-initiative refinement process to identify more clearly the user's problem state.

2. RELATED WORK
In recent work relating CBR to the Semantic Web [2, 4], we find the definition of two ontologies, CaseML and CBROnto. These are both defined for CBR rather then CCBR and thus do not define concepts related to question-answer (QA) pairs, which are at the core of the CCBR process. Nonetheless we considered these when we designed and implemented our OWL-based ontology, which we call CCBROnto (this has no relation to CBROnto). We make use of this ontology within our personalised service discovery and composition framework (PreDiCtS) to define cases of best practice composition knowledge. In what follows we make explanatory references to this ongoing work.

3. CCBRONTO
In CCBROnto the basic components of a *Case* are defined by the *CaseContext*, *Problem* and *Solution* classes. This structure is motivated by the underlying methodology used in PreDiCtS. In this framework we adapt the CCBR approach to help the user refine his query for a particular service request. The problem description is defined by a set of discriminating QA pairs, which characterize a particular solution. On the other hand, the solution is a place holder for a reusable service composition template which is a container of best practice knowledge about composition of generic service components. In the following sections we will explain in more detail the basic *Case* components and illustrate by means of an example how such a case is defined.

3.1 Context
In [3], the term context is defined as "*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*".

We fully agree with this definition and in the *CaseContext* we have included knowledge related to the case creator, case history, and case provenance. We have also considered ideas presented in [7] and [1] which discuss the importance of context in relation to Web Services. In PreDiCtS context knowledge helps to identify, *(i)* why a case was created and by whom, *(ii)* certain aspects of case usage and *(iii)* the case relevance to problem solving. The *CaseCreator* includes a reference to the *Role* description, that the creator associates himself with, together with a *foaf:Person* instance-definition that describes who this person is. The motivation behind using foaf is to keep

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 11th - 14th June, 2006

1This research has partially been funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Program project REWERSE number 506779

track of reputation knowledge which could be used to reliably share cases between PreDiCtS users.

```
<ccbr:Case rdf:ID="case1">
  <ccbr:CaseContext rdf:ID="cntxt1">
    <ccbr:hasProvenanceURI rdf:resource="http://www......org"/>
    <ccbr:hasCaseCreator>
      <ccbr:CaseCreator rdf:ID="ccr1">
        <ccbr:hasRole rdf:resource="&role;#KnowledgeEng"/>
        <foaf:Person>
          <foaf:name>Joe Black</foaf:name>
          <foaf:mbox rdf:resource="mailto:joe@test.org"/>
          <foaf:homepage rdf:resource="http://www...../joe"/>
        </foaf:Person>
      </ccbr:CaseCreator>
    </ccbr:hasCaseCreator>
  </ccbr:CaseContext>
  <ccbr:Problem rdf:ID="prob1">
    <ccbr:QAPairList>
      <list:first>
        <ccbr:QAPair rdf:nodeID="quest1"/>
      </list:first>
      <list:rest rdf:resource="&list;#nil"/>
    </ccbr:QAPairList>
  </ccbr:Problem>
  <ccbr:Solution rdf:ID="sol2">
    <ccbr:hasAction>
      <ccbr:OWLSTemplate rdf:ID="tmpl3">
        <ccbr:hasServiceTemplate rdf:resource="#Trav_Serv"/>
        <ccbr:hasProcessTemplate rdf:resource="#Trav_Proc"/>
        <ccbr:hasProfileTemplate rdf:resource="#Trav_Prof"/>
      </ccbr:OWLSTemplate>
    </ccbr:hasAction>
  </ccbr:Solution>
</ccbr:Case>
```

Figure 1: CCBROnto Case instance definition

The *CaseContext* also provides a place holder for *CaseHistory*, which becomes important when it comes to case ranking and usage, since it allows users to identify the relevance and usefulness of a case in solving a particular problem. It is also important for the case administrator when case maintenance is performed. Cases whose history indicates negative feedback may be removed from the case base. Case *Provenance* is also used in conjunction with reputation issues, since it associates a case with a URL indicating the case-origin.

3.2 Problem

The *Problem* state description in a PreDiCtS case is based on the taxonomic theory of [6]. Every problem is described by a list of QA pairs rather than a bag. This is required since QA pairs have to be ranked when they are presented to the user. Each QA pair consists of a *CategoryName*, a *Question* and an *Answer*. Since the taxonomic theory requires that QA pairs are defined in a taxonomy during the case creation stage, each question description is associated, through the property *isRelatedTo*, with an ontological concept defined in the domain of discourse. This relation is not intended to fully capture the natural semantics of the QAs, rather it is important when calculating similarities.

A typical QA pair example from the traveling domain might include the question, "*What type of transportation?* This is related, by means of the *isRelatedTo* property, to the concept *Transportation*, which is defined in the Traveling domain. On the other hand, we assume that *Answers* could have either a binary or nominal value and are respectively defined in the ontology by the *YesNoAnswer* and *ConceptAnswer* classes. The former points to the binary literals, while the latter is used to represent answers that are associated to a concept in a domain ontology through the previously mentioned *isRelatedTo* property.

3.3 Solution

The solution in PreDiCtS provides a hook where composition templates can be inserted. Each *Solution* is defined to be an *Action* which has a description and a composition template. A template can be sub-classed by a description such as that defined by OWL-S, as shown in Figure 1, though in practice it can be specialized also by other service descriptions.

4. CONCLUSION

Through the use of CCBROnto we are able to define cases whose solutions are composition templates. This allows our PreDiCtS framework to retrieve such templates by consulting the user in every stage and presenting her with the most suitable composition knowledge available to choose from. The user can then decide whether to reuse as is, or possibly adapt this to fit her personal needs.

5. REFERENCES

[1] Bry, F., et al, *Context Modeling in OWL for Smart Buildings*, in proceedings of GvB2005.

[2] Chen, H., et al, *CaseML: RDF based markup language for CBR on the Semantic Web*, ICCBR 2003

[3] Dey, A.K., *Understanding and Using Context*, in proceedings of Personal and Ubiquitous Computing, issue on Situated Interaction and Ubiquitous Computing, February 2001.

[4] Diaz-Agudo, B., et al, *On Developing a Distributed CBR Framework through Semantic Web Services*, OWL Experiences and Directions Workshop, Galway, 2005

[5] Goderis, A., et al, *Seven bottlenecks to workflow reuse and repurposing*, 4th Int. Semantic Web Conf., Galway, Ireland, 6-10 Nov. 2005

[6] Gupta, K., *Taxonomic Conversational Case-Based Reasoning*, 4th International Conference on CBR, 2001

[7] Maamar, Z., et al, *Context for Personalized Web Services*, 38th Hawaii International Conference on system Science, 2005

[8] Rajasekaran, P., et al, *Enhancing Web services description and discovery to facilitate composition,* First International Workshop, SWSWPC, July 2004

[9] Sirin, E., Parsia, B., et al, *Filtering and selecting semantic web services with interactive composition techniques*, IEEE Intelligent Systems, 19(4): 42-49,2004

[10] Sirin, E., et al, *Planning for web service composition using SHOP2*, Journal of Web Semantics, 1(4):377-396, 2004

[11] Sirin, E., et al, *Template-based composition of semantic web services*, AAAI Fall Symposium on Agents and the Semantic Web, Virginia, November 2005.

[12] Weber, B., et al, *CCBR-Driven Business Process Evolution*, Proc. 6th Int. Conf. on Case-Based Reasoning (ICCBR'05), Chicago, August 2005

From BPEL4WS Process Model to Full OWL-S Ontology

Muhammad Ahtisham Aslam
Betriebliche Informationssysteme
University of Leipzig
Germany
+49 (341) 97-32323
aslam@informatik.uni-leipzig.de

Sören Auer
Betriebliche Informationssysteme
University of Leipzig
Germany
+1 (215) 898-2716
auer@seas.upenn.edu

Jun Shen
School of IT and CS
University of Wollongong
Australia
+61-2-42213873
jshen@uow.edu.au

ABSTRACT

BPEL4WS is one of the most utilized business process development languages. It can be used to develop executable business processes as a combination of Web Services interactions in a specific sequence called process flow. But still BPEL4WS lacks sufficient representation of business process semantics required for business processes automation. On the other hand OWL-S (OWL for Web Services) is designed to present such kind of semantic information. There exists similarity in the conceptual model of OWL-S and BPEL4WS that can be used to overcome this lack of semantics in BEPL4WS by mapping the BPEL4WS process model to the OWL-S ontology. The mapped OWL-S service can be dynamically discovered, composed and invoked on the basis of matching semantics. Such a process of mapping syntax based Web Services composition in the form of BPEL process model to Semantic Web Services composition in the form of OWL-S composite service can also enable automation of BPEL processes as OWL-S services by applying AI planning techniques. In this paper we present a mapping strategy and a mapping tool that can be used to map BPEL processes to the OWL-S suite of ontologies.

Keywords

Web Services, Semantic Web Services, Business Process Execution Language for Web Services.

1. INTRODUCTION

Different workflow languages specially Business Process Execution Language for Web Services (BPEL4WS) [1] use Web Services in a more meaningful way by combining Web Services functionality in a specific sequence to perform a certain task. Even though BPEL has good process modeling capabilities, its semantic limitations are a hurdle in business process automation. OWL-S (OWL ontology for Web Services) [2], aims to make Web Services descriptions more computer-interpretable, thus enabling automation of a variety of tasks including Web Service discovery, invocation, and composition. Therefore, mapping a BPEL process to an OWL-S service can help to automate business processes on the basis of semantic information provided by the OWL-S ontology.

Our work (an improvement and extension to [3]) presents a mapping strategy and its prototypical implementation as mapping tool[1] (BPEL4WS2OWL-S) that can be used to map BEPL4WS processes to the complete OWL-S suite of ontologies. The whole mapping process uses the OWL-S API [4] on its backend for writing the OWL-S ontology for resulting OWL-S Service.

2. Mapping Specifications

BPEL has two kinds of activities, primitive activities and structured activities. BPEL primitive activities are mapped to the OWL-S *Perform* control construct to perform the relevant atomic process. Also, if a primitive activity is an input/output (I/O) activity (working as BPEL process interface) then this activity is used to create the *Profile* of the resulting OWL-S service. BPEL structured activities are mapped to relevant OWL-S control constructs as shown in figure 1.

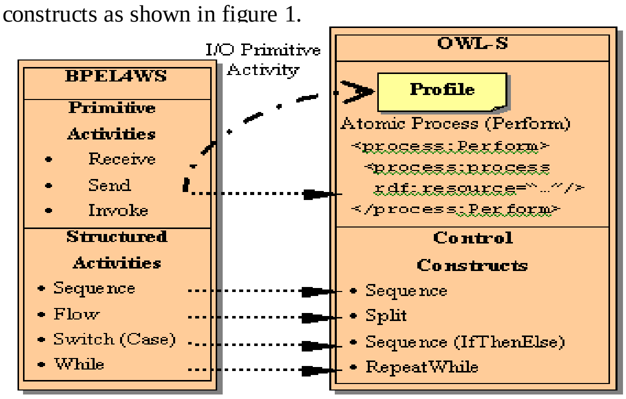


Fig.1. Overview of mapping specifications.

- **BPEL Process Mapping to OWL-S Process Model:** BPEL executable processes are mapped to OWL-S atomic and composite processes within the *Process Model* ontology. Also, to keep the mapping complexities within limitations synchronization between process components is not supported in this version.
- **Atomic Processes:** Operations supported by partner services (WSDL services) can be used to perform some specific task in a single step. Like an operation supported by a Web Service, an atomic process in OWL-S is a process that can perform some action in a single step. Therefore partner Web

[1] http://bpel4ws2owls.sourceforge.net/

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 11th – 14th June, 2006.

Services (WSDL Services) are parsed and an atomic process (with *Profile, Process Model and Grounding*) is created for each supported operation.

- **Primitive Activities and Atomic Processes:** The *Perform* control construct in OWL-S is used to perform an atomic process, while BPEL has primitive activities (e.g. *Receive, Invoke and Reply*) that can be used to perform some specific operation in a single step. Therefore, BPEL primitive activities are mapped to OWL-S *Perform* control construct to perform the relevant atomic process.
- **Structured Activities:** BPEL structured activities are mapped to OWL-S control constructs within an OWL-S composite process. Fig.1 shows the mapping of BPEL structured activities to OWL-S control constructs.
- **Data Flow:** Mapping of assignment activity that is used between two primitive activities, results in the creation of data flow between the corresponding atomic processes.
- **Profile:** A BPEL process can have one or more primitive activities, which act as an interface to communicate with the BPEL process. Therefore, among these primitive activities options, the input message of the first *Receive* primitive activity receiving a message from the outer world is defined as an input for the OWL-S composite process. If a *Receive* activity has corresponding *Reply* activity then the message variable of this *Reply* activity is used to set the output of the OWL-S composite process. If a *Receive* activity has no corresponding *Reply* activity then, the first primitive activity (e.g. the first *Invoke* activity sending some message to outer world) is taken as an output activity to define the output of the OWL-S composite process. Also, a primitive activity is declared as an Input/Output (I/O) activity if the BPEL's corresponding WSDL file supports its port type and operation. These input and output messages are used to create the *Profile* of the resulting OWL-S service. This *Profile* is used to present the semantically enriched service capabilities by annotating input and output parameters of *Profile* with ontological concepts.
- **Grounding:** The grounding of the mapped OWL-S service specifies the location of the grounding of each atomic process (created during the mapping as discussed above). Of course, the mapping is not able to define the XSL transformation for complex messages. Web Services Description Language (WSDL) service, being an XML format for describing network services is described in the grounding of each atomic process to have access to the original implementation of the WSDL service.

3. User Interface

The BPEL4WS2OWL-S mapping tool provides an easy to use interface (fig.2) employing menus and buttons to perform the mapping process. The mapping process includes creating a new project, adding input BPEL and WSDL files, validating the input files, building the project (to create object view of input files) and finally mapping the project. The resulting OWL-S ontology files can be viewed in the project explorer (upper right window) and the contents of these files can be seen in upper left window of the tool. The lower left window acts as an output window to see the output of different mapping actions. The lower right window is an object explorer, which gives an object view of the input files.

4. Conclusion and Future Work

OWL-S is not as mature as BPEL. For example, equivalents of BPEL activities like *Assignment*, *Fault Handler*, *terminate* etc. are not available in OWL-S for direct mapping from BPEL to OWL-S. Also, users need to annotate the *Profile* of the resulting OWL-S service with their domain ontologies. Therefore, manual changes are required in the areas where the mapping is only partially supported or information needs to be added by the user. Such manual changes are a time consuming and complex task and require a user to be an expert of OWL-S. So at this stage our BPEL4WS2OWL-S tool needs constant updates with the upcoming versions of the related technologies. Furthermore, a tool is needed that can be used to develop domain ontologies and an editor which helps in editing the resulting OWL-S ontology with these domain ontologies more easily, ideally in a visual environment. Protégé with its plugin, OWL-S Editor, is an ideal basis to achieve this goal. Hence, we are working to improve our tool, and to make it available as a BPEL4WS2OWL-S import plug-in for Protégé with OWL-S Editor, so that the mapped OWL-S services can be directly imported in the OWL-S Editor tab and can thus be edited in a visual environment.

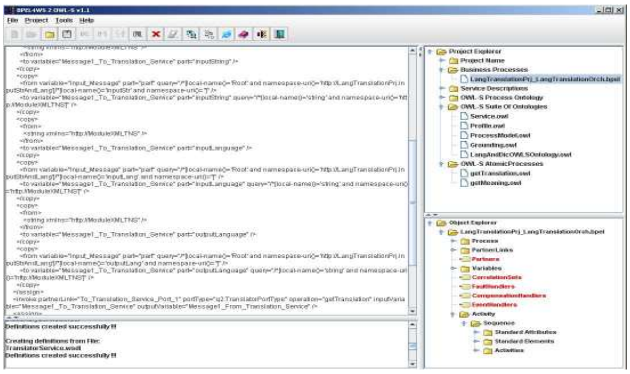


Fig.2. Overview of BPEL4WS2OWL-S mapping tool interface.

5. REFERNECES

[1] Business Process Execution Language for Web Services Version 1.1. 5th May 2003. [online] Available ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf.

[2] OWL-S: Semantic Markup for Web Services. [online] Available http://www.daml.org/services/owl-s/1.1/overview/.

[3] J. Shen, Y. Yang, C. Zhu and C. Wan. "From BPEL4WS to OWL-S: Integrating E-Business Process Descriptions", *Proc. of 2nd IEEE International Conference on Services Computing (SCC 2005)*, pp.181-188, Orlando, USA, July 2005.

[4] OWL-S API : [online] Available http://www.mindswap.org/2004/owl-s/api/

Towards an Ontology Metadata Standard

Raul Palma
Facultad de Informática. Universidad Politécnica de Madrid
Campus Montegancedo, s/n 28860
Boadilla del Monte. Madrid. Spain
34-913367439
rpalma@fi.upm.es

Jens Hartmann
Institute AIFB
University of Karlsruhe
Germany
49-721 608 6554
hartmann@aifb.uni-karlsruhe.de

Asunción Gómez-Pérez[i]
Facultad de Informática. Universidad Politécnica de Madrid
Campus Montegancedo, s/n 28860
Boadilla del Monte. Madrid. Spain
34-913367439
asun@fi.upm.es

ABSTRACT

In this poster, we present (i) a proposal for a metadata standard, known as Ontology Metadata Vocabulary (OMV) which is based on discussions in the EU IST thematic network of excellence Knowledge Web[1] and (ii) two complementary reference implementations which show the benefit of such a standard in decentralized and centralized scenarios, i.e. the Oyster P2P system and the Onthology metadata portal.

Categories and Subject Descriptors

I.2.4 [**Knowledge Representation Formalisms and Methods**]: Representation languages.

K.6.4 [**System Management**]: Centralization/descentralization

General Terms

Management, documentation, design, reliability, experimentation, standardization.

Keywords

Ontology, Metadata, Peer-to-Peer, Repository

1. INTRODUCTION

Ontologies have undergone an enormous development and application in many domains within the last years, especially in the context of the Semantic Web. Currently however, efficient knowledge sharing and reuse, a pre-requisite for the realization of the Semantic Web vision, is a difficult task since it is hard to find and share existing ontologies because of the lack of standards for documenting and annotating ontologies with metadata information. Without an ontology-specific metadata developers are not able to exploit existing ontologies, which leads to problems of interoperability as well as duplicate efforts. Then, in order to provide a basis for an effective access and exchange of ontologies across the web it is necessary to agree on a standard for ontology metadata, that is a common set of terms and definitions describing ontologies, that is called metadata vocabulary. Furthermore, an appropriate technology infrastructure is required, e.g. tools and metadata repositories, compatible to the ontology metadata standard, must be developed to support the creation, maintenance and distribution of ontology metadata.

2. OMV

Some of the aspects captured by OMV[2] (the complete ontology is described in [1]) are similar to other metadata standards, like Dublin Core [2]. However, some important differences like the conceptual models (semantics) behind ontologies require a detailed analysis and a different representation of metadata about ontologies. From a conceptual design point of view, OMV distinguishes between the OMV Core, which captures information relevant to the majority of ontology reuse settings and various OMV Extensions that allow ontology developers/users to specify task/application-specific ontology-related information.

2.1 Overview

OMV core distinguishes between an ontology conceptualisation and its implementation(s) in concrete representation languages. From an ontology engineering perspective, a person first develops such core idea of what should be modeled (and maybe how) in his mind. Further, this initial conceptualisation might be discussed with other persons and then, an ontology will be built using an ontology editor and stored in a specific format. Over time, several realizations of this initial conceptualisation might be created in many different formats, e.g. in RDF(S) or OWL. The two concepts are defined as follows:

Ontology Conceptualisation: *(OC)* represents the (abstract) core model or idea behind an ontology. It describes the core properties of an ontology, independently of any implementation details.

Ontology Implementation: An *(OI)* represents a specific realization of a conceptualisation. It describes properties of an ontology that are related to the realization or implementation.

The distinction between the two concepts provides an efficient mechanism for the realization of several ontology management utilities, such as the tracking of several versions, the evolution flow of an ontology or the handling of different representations of the same knowledge model. OMV also models additional classes that are required to represent and support the reuse of ontologies by such metadata vocabulary, especially in the context of the Semantic Web. Hence, we modeled further classes and properties representing *environmental information* and *relations* such as: Party, Organisation, Person, OntologyType, LicenseModel, OntologyLanguage, etc. The main classes and properties of the OMV ontology are illustrated in Figure 1.

3. USE CASES

We shortly introduce two complementary applications based on OMV, namely the decentralised P2P system Oyster[3] and the centralized metadata portal Onthology[4], to show the benefits of using such a vocabulary in real life scenarios. Both applications

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva Montenegro, 11th - 14th June, 2006

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

[2] OMV ontology is available at http://ontoware.org/projects/omv/

[3] Available at http://oyster.ontoware.org/

[4] http://www.onthology.org/

have in common that they support single users and

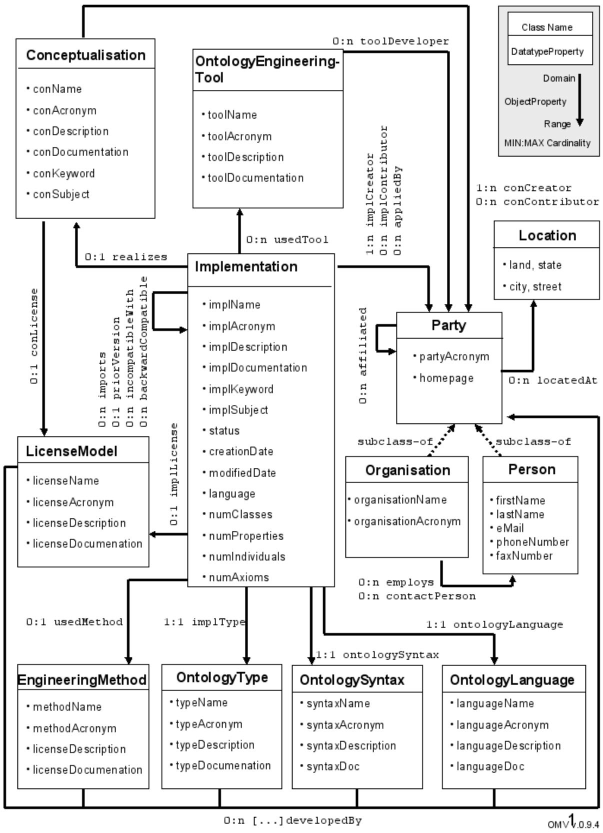


Figure 1. OMV overview

communities of users in *identifying*, *reusing* and *providing* ontology metadata. However, both applications are covering a variety of different tasks and have different usage perspective. For users who want to store metadata individually, a repository is required to which the user has full access and can perform any operation without any consequences to other users. In this situation a decentralised system is the technique of choice, as it allows the maximum of individuality while it still ensures exchange with other users. Centralized systems allow reflecting long-term community processes in which some ontologies become well accepted for a domain or community and others become less important. The benefit of connecting both systems lies mainly in the simple use of ontology metadata information existing within Oyster. So, while users are applying or even developing their own ontologies they can manage their own metadata along with other existing metadata in Oyster. If some metadata entries from Oyster have reached a certain confidence, they can be easily imported into Onthology.

4. RELATED WORK

We will briefly mention related metadata standards, in particular those relevant to the Semantic Web. The **Dublin Core (DC)** metadata standard [2] is a simple yet effective element set for describing a wide range of networked resources. The **Reference Ontology** [3] is a domain ontology that gathers, describes and links existing ontologies. However its focus is to characterize ontologies from the user point of view, and provides only a list of property-value pairs for describing ontologies. **FOAF** [4] provides a way to create machine-readable Web homepages for people, groups, companies and other things. The Semantic Web search engine **SWOOGLE** [5] makes use of particularly metadata which can be extracted automatically. There exist some similar approaches to our proposed solution to share ontologies, but in general their scope is quite limited. E.g. the **DAML ontology library** [6] provides a catalog of DAML ontologies that can be browsed by different properties. The **FIPA ontology service** [7] defines an agent wrapper of open knowledge base connectivity. Finally the **SchemaWeb Directory** [8] is a repository for RDF schemas expressed in RDFS, OWL and DAML+OIL.

5. CONCLUSIONS AND FUTURE WORK

A key issue for sharing knowledge on the Semantic Web is to reuse existing ontologies. Our contribution aims at facilitating reuse of ontologies which was previously unknown for ontology developers by providing an Ontology Metadata Vocabulary (OMV) and two applications for decentralized (Oyster) and centralized (Onthology) sharing of ontology metadata based on OMV. Our current work is DEMO [9], a framework for the development and deployment of ontology metadata, which comprises OMV and an inventory of methods to collaboratively extend OMV in accordance to the requirements of an emerging community of users, and tools for metadata management. Finally, our future work includes many challenges such as the application of OMV extensions, the evaluation of the application of OMV in different scenarios and pushing OMV to a community standard.

6. ACKNOWLEDGMENTS

Our thanks to our partners from the EU project Knowledge Web for their present and future collaboration. Especially we would like to thank Elena Paslaru (FUB), Denny Vrandecic (AIFB).

7. REFERENCES

[1] J. Hartmann, R. Palma OMV-Ontology Metadata Vocabulary for the semantic web.2005.v1.0. http://omv.ontoware.org

[2] Dublin Core. http://dublincore.org.

[3] J. Arpirez et al. Reference Ontology and (ONTO)2 Agent: The Ontology Yellow Pages. Knowledge and Information Systems. Volume 2. 2000. 387-412.

[4] FOAF- "Friend Of A Friend"-. http://www.foaf-project.org

[5] Li Ding et al. Swoogle: A search and metadata engine for the semantic web. *In Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management*, pages 58–61, November 2004.

[6] DAML Ontology repository. http://www.daml.org/ontologies

[7] H. Suguri et al. Implementation of fipa ontology service. In *Proc. of the Workshop on Ontologies in Agent Systems, 5th Int. Conf. on Autonomous Agents Montreal, Canada*, 2001.

[8] SchemaWeb directory. http://www.schemaweb.info/

[9] J. Hartmann et al. DEMO-A Design Environment for Metadata Ontologies. In proc. of ESWC2006. June 2006

---

[i] Additional Authors: York Sure (AIFB), M. Carmen Suarez-Figueroa (UPM), Peter Haase (AIFB) and Rudi Studer (AIFB).

A Personalized Hotel Selection Engine

Jörg Garbers
Freie Universität Berlin
Institut für Produktion,
Wirtschaftsinformatik und OR
Garystr. 21, D-14195 Berlin
jgarbers@zedat.fu-berlin.de

Magnus Niemann
Freie Universität Berlin
Netzbasierte
Informationssysteme
Takustr. 9, D-14195 Berlin
maggi@inf.fu-berlin.de

Malgorzata Mochol
Freie Universität Berlin
Netzbasierte
Informationssysteme
Takustr. 9, D-14195 Berlin
mochol@inf.fu-berlin.de

ABSTRACT
In this poster we describe the results of our work in the *Reisewissen* project which adopts Semantic Web (SW) technologies for the tourism domain. In the prototypical implementation of a SW based portal for hotel booking we use RDF as a shared data representation scheme utilized by evaluation components written in Java and Prolog. We apply Prolog to transform data between external and internal ontologies and to represent expert knowledge. Since our prototypical implementation is to be integrated into a real world production system where performance plays a crucial role, the developed solution is based on a mix of SW technologies and classical knowledge representation tools.

Keywords
Tourism Domain, Semantic Web, Ontologies, OpenGuides

1. MOTIVATION
The past 10 years of Web evolution has led to the establishment of electronic markets. The next 10 years may be characterized by the transformation of the Web from a document publication medium intended for human utilization into a medium for intelligent knowledge exchange [3]. In the light of these developments the tourism domain, which is already exhibiting a gradual shift towards electronic transactions, can benefit from embracing new and emerging technologies.

Although current online travel systems support the customer in finding a suitable hotel or even a whole trip, most of the work is still up to the customer, who has to consider several sources of information (hotel review sites, booking portals, hotel websites) before deciding which hotel to book.

Our goal in the *Reisewissen* project (`reisewissen.ag-nbi.de`) is, on one hand, to support the user in the choice of a hotel by *selecting* and *ranking* suitable hotels and, on the other hand, to evaluate the usefulness of SW [1] technologies in this context.

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

The hotel selection and ranking is accomplished by the integration of several heterogeneous data sources into the hotel evaluation process and semantically matching the collected hotel information to the customer's individual profile. Furthermore, we enrich the collected data with domain expert knowledge (collected during "experts' interviews") represented with Prolog rules on top of RDF(S)/OWL [4]. The developed selection and ranking engines will be used by our industrial partner, *ehotel AG* (`www.ehotel.de`) to enhance its online hotel booking system. Since our prototypical implementation is to be integrated into a real world production system where performance plays a crucial role, the final system is based on usage of a mix of SW technologies and classical knowledge representation tools.

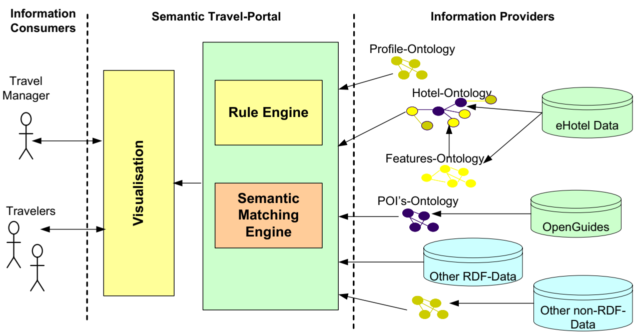


Figure 1: *Reisewissen* hotel recommendation engine

2. TECHNICAL REALIZATION
When we view our system (see Fig. 1) in a bottom-up way, we integrate several heterogeneous data sources, provide means to formalize and use expert knowledge and develop a suitable user interface for collecting customer requirements, expert knowledge and data from additional sources.

To rank hotels in a personalized way, we transform the customer's requirements into mathematical objective functions. A weighted combination of these functions is evaluated against each member of a set of hotels, yielding a ranking for the respective hotel (see Fig. 2).

2.1 Data Sources and Ontologies
We integrate various RDF and non-RDF data from different sources (see Fig. 1) such as hotel information (location, room rates, amenities etc.) provided by *ehotel AG* in XML

format according to the Open Travel specification (`www.opentravel.org`), hotel reviews (spreadsheets) and information about points of interest (POI) in RDF format from OpenGuides (`www.openguides.org`). External RDF data is transformed into our own domain specific ontology by means of *transformation rules* formalized in Prolog or Java while for proprietary, non-RDF data we provide an RDF view or RDF dump. Where performance is crucial, we keep or cache data in specialized data structures. For example, we extract location based RDF information from OpenGuides, translate it to our POI ontology and cache geocoded information in grid-like data structures for fast lookup.

2.2 Rules

To use concrete RDF data to estimate hotel's quality w.r.t specific customer's requirements we use Java methods that may query Prolog rules. The latter are used for two purposes in our project:

Rules for consolidating data into knowledge, e.g. the rule of thumb that everywhere in London there are Indian restaurants because of the high Indian population. Customers who require vegetarian food will be satisfied everywhere, without knowing about particular instances of vegetarian restaurants near a hotel.

Rules used in matching process to compare customer profile properties on predicated knowledge with the characteristic of a hotel, e.g. the physically disabled customer who is planing to use public transportation needs a hotel near a bus/subway station which also has access to an elevator.

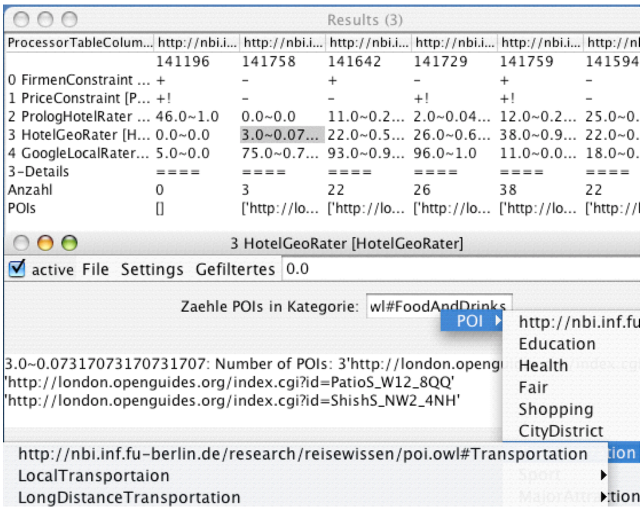


Figure 2: The domain expert's testing environment

2.3 (Expert) User interfaces

In order to develop an application which can support all types of prospective users (domain experts, end and power users) we implement different user interfaces.

Tourism domain experts need a workbench to test the quality of data and rules (see Fig. 2). A number of hotels (table headers) are evaluated against several objective functions (leftmost column in the table). A function is implemented as a java class that specifies the parametrization user interface in terms of text fields and choices. A function instance is parameterized by the expert. Text field values may be set by picking URIs from an ontology driven popup menu. The experts formulate basic evaluation functions and compare the ranking result with their expectations. They can also combine several basic evaluation functions into complex ones which can be chosen by the end user (e.g. suitability of the hotel for wellness, business). *Advanced users* may wish to build their own objective function by building a preference tree over given basic evaluation functions. Evaluation functions can be parameterized using a form-based interface and ontology based functions using a tree of weighted ontology concepts.

3. CONCLUSIONS

The requirements analysis of the hotel recommendation engine raised several constraints which we have to take into account. We have to ensure that the engine is flexible with regard to later adaption to the production system and efficient regarding the end user querying process. Furthermore it has to allow an easy integration of information sources and provide means to generate new information from appropriately formalized expert knowledge. The excess value of semantic web technologies lies in ontology-based applications like semantic matching and similarity searches, moreover we make use of already published RDF metadata. When it comes to querying at runtime and numerical evaluation, those technologies proved to be less efficient.

From the implementation point of view we found that using Java/Jena as a RDF framework was cumbersome and slow. For the most cases where data could be kept in memory, SWI prolog's Semantic Web Library (`www.swi-prolog.org/packages/semweb.html`) was the better choice, thus using Prolog as the rule language appeared reasonable. In the existing prototype Java/Swing is used for the user interface and for simple evaluation tasks. Prolog is connected via the java native interface. For system simplicity we consider to dump the prolog generated knowledge into SQL databases, from where the data could be provided as virtual Jena graphs via D2RQ[2].

4. ACKNOWLEDGMENTS

This work is part of the *Reisewissen* project, a cooperation between Freie Universität Berlin and ehotel AG; supported by the Investitionsbank Berlin (IBB) and by the European Regional Development Fund (ERDF).

5. ADDITIONAL AUTHORS

Additional authors: R. Oldakowski (`radol@gmx.de`), R. Tolksdorf (`tolk@inf.fu-berlin.de`) and C. Bizer (`bizer@zedat.fu-berlin.de`) – all Freie Universität Berlin.

6. REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, pages 34–43, May 01.

[2] C. Bizer and A. Seaborne. D2RQ – Treating Non-RDF Databases as Virtual RDF Graphs. *Proc. of the 3rd Int. Semantic Web Conference (ISWC2004)*, 2004.

[3] J. Hendler, T. Berners-Lee, and E. Miller. Integrating Applications on the Semantic Web. *Journal of the Institute of Electronic Engineers of Japan*, 122(10):676–680, Oct. 02.

[4] G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. http://www.w3.org/TR/rdf-concepts/, Feb. 04.

Paving the way towards the e-humanities: a Semantic Web approach to support the learning of philosophy

Michele Pasin, Enrico Motta
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA, UK
+44 (0) 1908 858730
{m.pasin, e.motta}@open.ac.uk

ABSTRACT
Use of the Semantic Web for e-Learning (SWEL)[1] brings great advantages by the accurate description of the semantics of a domain, in order to enhance the navigation and retrieval of the related resources. Even though much work has already been done in relation to scientific areas of research (biology, physics, computer science), in the humanities there has not been the same urgency of delivering the advantages of the new technologies to the classroom or scholars. In this paper, we sum up the research we are pursuing for an e-Learning approach to the field of philosophy, based on ontological engineering and narrative studies.

Categories and Subject Descriptors
K.3.1 [Computer Uses in Education] - Computer-assisted instruction (CAI), Computer-managed instruction (CMI), Distance learning. H.5.4 [Information Interfaces and Presentation]: Hypertext/Hypermedia - Architectures, Navigation, User issues. J.5 [Computer Applications – Arts and Humanities]: Fine arts.

General Terms
Design, Standardization.

Keywords
Ontology, humanities, philosophy, learning narratives.

1. INTRODUCTION
Recent research on the application of Semantic Web technologies to e-Learning (SWEL) has already produced various results [9]. Ontologies can be used to describe learning resources directly, or to provide a common ground on which to map LOs annotated using the traditional metadata standards. Ontologies are also used to describe other dimensions involved in the educational scenario (pedagogical assumptions, presentation strategies). It is not our purpose here to sum up these attempts; instead, we would like to highlight the fact that in contrast to much work done in scientific fields such as biology, physics or computer science (readers can take as a proof of this the various e-science projects [4]), there are very few e-Learning systems in the humanities that adopt knowledge representation techniques in order to enhance the usage and understanding of available digital artifacts. This, in spite of the great number of resources on the web, and the richness of these domains' semantic relations, translatable in non trivial browsing facilities. In the humanities knowledge is not usually as structured and hierarchically organized as it would be in computer science, for example. Here, or in any other "scientific" domain, in fact, the taxonomical relations between the concepts represented are often enough, in order to provide useful navigation structures [5].

In the following sections we describe in more detail our approach to the formalization of a specific domain in the humanities, philosophy. Section 2 introduces the ontology we have created to describe at a fine level the knowledge needed in the teaching of philosophy; section 3 deals with the model, drawn from narrative studies, we are using in order to support a constructivist approach to learning; section 4 concludes with a description of the ongoing and future work.

2. ONTOLOGY FOR PHILOSOPHY
Within the PhiloSURFical project[2] we are defining an ontology that captures the various dimensions involved in the philosophical work. The ontology, being engineered with a clear educational purpose in mind, could be divided into three super-categories: the empirical domain, the pedagogical domain, and the theoretical domain of a philosophical resource. The empirical domain is used to describe all the knowledge related to the material and not-domain-specific aspects of a philosophical resource, such as *authors*, *dates*, *places* etc. In doing so, we have readapted and extended the AKT reference ontology [1]. The pedagogical domain abstracts the educational value of a resource, its role in the overall structure from the educational point of view.

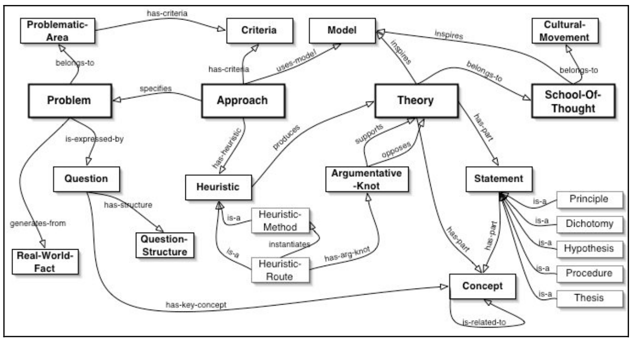


Fig.1 Theoretical domain of the philosophical ontology.

A previous and valuable attempt to model this dimension is the one done by Ullrich [11], that we have used as a starting point.

[1] Consider for example the series of International Workshop on Applications of Semantic Web Technologies for E-Learning, http://www.win.tue.nl/SW-EL/2005/index.html.

[2] www.kmi.open.ac.uk/people/mikele/philontology/main.htm

The theoretical domain, finally, tries to reflect the way things happen in the philosophical work. It models philosophical concepts such as *problem, school-of-thought, approach, theory* so as to emphasize their interdependence and their changes in meaning depending on the context.

This framework lets us go, for example, from Democritus to Russell, for they share a similar "interest" in atoms (although one in a physical sense, the other in a logical one) or from this latter to Popper, because beyond their involvement in epistemology they are also contemporaries and fierce opponents of the second world war. The first pathway mentioned is clearly a theoretical one, rooted in the contents of the authors' doctrines, while the second one is also historical, since it is based on the fact that the two authors lived at the same time.

All these dimensions are implemented using OWL and Protégé [10]. Even if here these three knowledge domains have been treated as separated for explanation purposes, they are instead grouped into the same ontology, which has been modeled taking inspiration from SUMO [7]. A version of the ontology is available online on www.kmi.open.ac.uk/people/mikele/philontology/Philo-know.owl.

3. LEARNING THROUGH STORIES

As already discussed elsewhere [8], we have adopted an approach to learning based on story construction. Within a classroom scenario, a teacher annotates his/her materials using the ontology, then lets students play with them through the medium of a story construction interface. This means that students can identify items of interest and use them as concepts in a *story*; they can declare the kind of *discourse* they would like these items to be connected with and the kind of *plot* the final hypertextual narrative should have. An adequate mapping from these classic narrative concepts [3] to knowledge base queries allows the dynamic reconstruction of the annotated materials into a personalized learning hypertext. Such a *learning narrative* can be, for example, the instance of a geo-historical discourse, of a theoretical one or probably, more often, of a mix of the two. So, for example, we can retrieve the different answers (*theories*) to the mind-body *problem* during the eighteenth century (*history*) in Europe and Asia (*geography*).

Moreover, these narratives result from the intersection of the domain semantics with the pedagogical one, adding a further dimension to the plot construction process. The reconstruction of Bloom's taxonomy of learning objectives [2] in the form of specific ways to traverse the semantic space is one of our goals.

This approach has already been tested within our department [6], although with less emphasis on the learning dimension and a more constrained application domain. Thus, we are working on implementing this framework in an extended and improved manner. From our first results, it is clear that the main axis (that is, the concepts' relations) needed to build valuable browsing facilities are not specific to the philosophical domain only. We therefore envisage other humanities' related domains where this approach could be replicated or extended. A desired outcome is also the definition of an abstract *learning narratives* ontology.

For example, going beyond the specific domain of philosophy, from a resource about Plato's theory of ideas it could be possible to browse, according to a specific learning narrative (*historical-context*, for example), to a document discussing the contemporary Peloponnesian war, or (following a more *conceptual* narrative) to a resource examining Raffaello's painting about the Athen's school. This last phase would end with the production of a series of reusable cross-domain semantic models for navigation.

4. CONCLUSION

This work is has been funded by the European Commission 6th Framework Program under the Knowledge Web project, and is now conducted in collaboration with the Department of Philosophy of the Open University. Two of their courses are being annotated using this ontology, and the resultant material will be used to experiment the creation of personalized learning narratives for the students. The same students and teachers of the Open University will be an ideal test bed for the final application, and the main source of data for the evaluation phase

5. REFERENCES

[1] AKT, *AKT Reference Ontology v.2 - AKTive Portal Ontology v.2*. 2002, http://d3e.open.ac.uk/akt/2002/portal-ocml-v2.0/portal-ocml-v2.0-t.html.

[2] Bloom, B.S., *Taxonomy of Educational Objectives - The classification of educational goals*, ed. Bloom, B.S. 1956: Longman.

[3] Chatman, S., *Story and Discourse*. 1978: Cornell University Press.

[4] Globe, C., *Hatching the next generation Web in the incubator of e-Science Or Lets stop doing stuff in the abstract, we nearly always get it wrong anyhow*. 2005, http://www.semanticgrid.org/docs/goblewebscipos.html.

[5] Henze, N. and Herrlich, M. *The Personal Reader: A Framework for Enabling Personalization Services on the Semantic Web*. in AH 2004: International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems. 2004.

[6] Mulholland, P. and Collins, T. *Using Digital Narratives to Support the Collaborative Learning and Exploration of Cultural Heritage*. in IEEE International workshop on Presenting and Exploring Heritage on the Web (PEH'02), DEXA 2002. 2002. Aix-En-Provence, France.

[7] Niles, I. and Pease, A. *Towards a standard upper ontology*. in Formal Ontology in Information Systems. 2001.

[8] Pasin, M. and Motta, E. *Semantic Learning Narratives*. in International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL), KCAP-05. 2005. Banff, Canada.

[9] Pasin, M., *Semantic Learning Narratives*, Technical Report KMI-05-07, 2005, Knowledge Media Institute.

[10] Stanford-Medical-Informatics, *The Protege Ontology Editor and Knowledge Acquisition System*, http://protege.stanford.edu/.

[11] Ullrich, C. *Description of an instructional ontology and its application in web services for education*. in Proceedings of Workshop on Applications of Semantic Web Technologies for E-learning. 2004. Hiroshima, Japan.

SEA: Introducing the Semantic Exchange Architecture

Thomas Franz, Carsten Saathoff, Olaf Görlitz, Christoph Ringelstein, Steffen Staab
ISWeb Group, Institute for Computer Science
University of Koblenz, Germany
http://isweb.uni-koblenz.de
{franz,saathoff,goerlitz,cringel,staab}@uni-koblenz.de

1. INTRODUCTION

With novel centralized information management systems[1], users benefit from collective data organization in the form of collective information tagging. The centralized architecture of these systems, however, imposes privacy and availability constraints as personal information needs to be handed over to a third party and offline utilization is impossible. The Semantic Exchange Architecture (SEA[2,3]) addresses these issues enabling autonomy of data management and freedom of data organization. SEA ensures privacy and fine grained access control through local data storage without lacking the advantages gained by collective information organization. Distributed information management is achieved through Peer-to-Peer networking in combination with semantic web technologies, while the benefits of collective information tagging are available through anonymous tag distribution in the Peer-to-Peer network. Information can be shared both publicly with all connected peers as well as privately within networks of trust.

As a use case example for SEA, we consider project work in which multiple institutions take part so that confidential data is distributed at multiple locations. The setup of a centralized information management system to share that information increases costs and decreases ease-of-use due to the typical upload and download procedures in centralized systems. SEA enables working group members to easily share their information, e.g. by tagging all relevant information with the name of the working group, and associating group access with that tag.

2. ARCHITECTURE

SEA constitutes a network of decentralized repositories in which information is collectively organized by tags. A repository runs on a local desktop and provides locally stored information as well as a portion of globally shared information. Taggings are used to enable local as well as networked access and exchange of the information distributed over multiple peers.

2.1 Organization of Information

Conventional hierarchical data organization is unable to represent different perspectives onto some data. SEA employs tagging, the association of user defined catchwords with information objects, as a mechanism to allow more flexible data management and retrieval. In contrast to taxonomies, tagging provides more freedom to organize information since it does not impose any relations between tags. An information object can be tagged with multiple tags to represent different perspectives onto the object.

Based on the assumption that multiple users associate the same meaning to a tag, sharing taggings allows further exploitations: First, by requesting all information objects tagged with a tag k, users can retrieve information objects related to k that they are not aware of, however have been tagged with k by other users. Second, users can find information objects that are related to an object o by requesting information objects that share one ore more tags with o.

2.2 Data Model

SEA employs ontologies as meta models (micro models[4]) for the managed data to achieve interoperability and extensibility. We further argue that building novel systems on ontologies from the beginning leverages integration of knowledge, reasoning and further improvements later on. Figure 1 illustrates how we combined ontologies that model tagging, information resources, and access control.

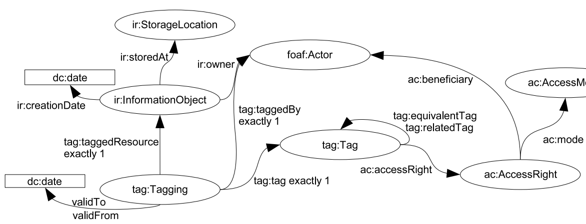


Figure 1: Ontologies in SEA

2.3 Access Control

[1]http://del.icio.us, http://www.flickr.com

[2]http://isweb.uni-koblenz.de/Research/sea

[3] This research was partially supported by the European Commission under contract FP6-001765, aceMedia. The expressed content is the view of the authors but not necessarily the view of the aceMedia project as a whole.

[4]http://esw.w3.org/topic/MicroModels

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

SEA utililizes tags to organize information but also to define access rights for the shared information. This approach is easy-to-use by users as only the system of tagging needs to be learned in order to both flexibly organize information and maintain access control. Access control is realized based on rules which define the behavior of SEA for non-trivial cases, e.g. if access rights are associated with multiple tags that classify the same information object, or one user associates access rights to a tag, while another user adds the same tag to an information object.

2.4 Data Distribution

Which data is distributed, and how the information retrieval of that data is implemented in SEA depends on how data is shared. We distinguish between public data (shared with everybody), and protected data (shared only with dedicated users). For public data, taggings and object locations are distributed to enable a simple object retrieval as well as exploitations as listed in 2.1. Dealing with protected data is more complex and explained in the following.

2.4.1 Combining Privacy and Collective Tagging

Obeying privacy demands and exploiting collective tagging are contradicting goals as privacy demands that data is not publicly shared while exploiting collective tagging demands to share information. SEA supports those exploitations whithout breaking privacy rules by only distributing anonymized taggings (identifiers of information objects and associated tags) for secured data. Such a distribution of taggings allows the identification of information objects by exploitations as listed in 2.1, however, due to the missing location information disallows their retrieval. The retrieval of protected data is based on the consultation of a finite list of peers to which the retrieving user is known, similar to a buddy list in instant messaging software. Consulted peers check whether the retrieving peer has appropriate access rights before providing the requested information. We argue that this solution is sufficient to find information objects that can be accessed by the particular user as owners of protected information objects are expected to know the users for which they grant access and vice versa. If one wants to grant access to users one does not know, public access can be granted.

2.4.2 Distribution Mechanism

SEA utilizes a distributed hashtable (DHT) approach to distribute information in the network. Four hashtables are employed to efficiently represent the needed information. Table tab_o contains for each information object id the set of all tags associated with that object and thus allows to retrieve all tags associated to an information object. Another table tab_k allows for querying in the opposite direction, i.e. retrieval of all object ids for a tag. While the computation of tag correlations would be possible by using only tab_o and tab_k, it would require multiple request and thus increase network load. We argue that memory and space costs are lower than those for network bandwith and model tag correlations by an additional DHT tab_{co} that maps each tag k to the set of tags that occur together with k. As we distribute location information for those information objects that are public, that information is contained by the table tab_l that maintains for each information object a set of locations where it is available.

3. IMPLEMENTATION

The main components of SEA are the data repository, SEA core, the peer manager, and the DHT module as depicted in figure 2. The repository is a RDF[5] store that supports

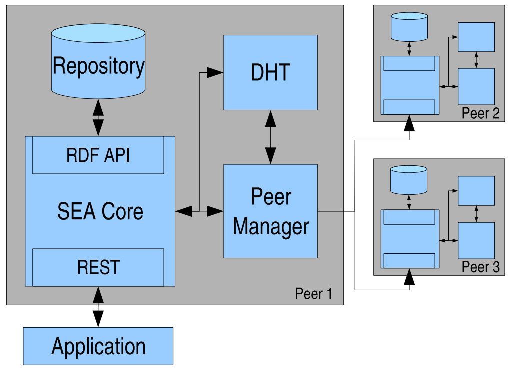


Figure 2: Implementation of SEA

SPARQL[6] and stores all metadata available in SEA. SEA Core constitutes the module which is directly accessed by applications using SEA and therefore is responsible for distributing all requests to appropriate modules. The interface of the core is a REST Api, that offers a number of central services: i) Requesting resources with a specific tag, tags of a resource with a specific identifier and tags that are related to another tag, ii) modifying the tags of resources, and iii) authentication of trusted users. The peer manager module is responsible for all operations involving the communication with other peers, i.e. rendevous, authentication and request forwarding. Additionally, it provides peer information for the DHT implementation. Communication with other peers is established via the common interface exposed by every peer. Results of forwarded requests are handed back to the SEA Core for further processing. The DHT module uses a distibuted hashtable implementation to efficiently store general tagging information so that it is available for all peers in the network. SEA is under development[7], efforts are concentrating on the SEA Core implementation and the integration with the Sesame2[8] RDF repository. Additionally, we started implementing a simple file browser that allows to tag arbitrary resources and submits taggings and other information to SEA Core. In parallel we also evaluate possible solutions for the DHT implementation, namely Pastry and Bamboo[9].

4. CONCLUSION

SEA tackles shortcomings of conventional information sharing platforms by providing secure, collective, and distributed information organization. SEA's open architecture offers easy adoption, extension, and development as it is based on acknowledged standards that are well supported by programming libraries and development tools. Work on SEA contributes to research on P2P systems, Social Network Analysis, and the Semantic Web (in particular the development of the Networked Semantic Desktop).

[5]http://www.w3.org/TR/rdf-primer/
[6]http://www.w3.org/TR/rdf-sparql-query/
[7]http://isweb.uni-koblenz.de/Research/sea
[8]http://openrdf.org
[9]http://bamboo-dht.org/

Task Ontology-Based Framework for Modeling Users' Activities for Mobile Service Navigation

Munehiko Sasajima[1], Yoshinobu Kitamura[1], Takefumi Naganuma[2], Shoji Kurakake[2] and Riichiro Mizoguchi[1]

[1]I.S.I.R., Osaka University
8-1 Mihogaoka, Ibaraki,
Osaka, 567-0047 Japan
{msasa, kita, miz}@ei.sanken.osaka-u.ac.jp

[2]Network Laboratories, NTT DoCoMo, Inc.
3-5, Hikari-no-oka, Yokosuka City,
Kanagawa, 239-8536
{naganuma, kurakake}@netlab.nttdocomo.co.jp

Keywords
Task ontology, mobile service, model of users' activity

1. INTRODUCTION

We can get many kinds of mobile services via mobile handsets in Japan. According to an annual report by NTT DoCoMo [1], Japan's premier mobile communications company which manages mobile internet services and occupies about 58% of the market, we have more than 89,000 service sites today. On the other hand, a large number of services cause difficulties in searching, finding and selecting suitable services for consumer's needs.

One of the reasons for the difficulties is that menus of the current mobile services are organized from the viewpoint of the domain. Users have to learn the menu system to access the services; hierarchical structure of the menu, relation between name of the category and services in the category. If a user wants to catch the last train, for example, such a domain-oriented menu will guide him/her as follows: "menu", "latest information", "traffic", "train information", "timetable" and "input start station name". The user follows the menu and finally reaches the service that provides information about timetable. As this example shows, users have to translate "what they want to do" to "name of the menu" before getting mobile services they want.

On the other hand, there is another type of menu which we call "task-oriented menu" [2]. The goal of this research is to realize a task-oriented menu system which enables more efficient mobile

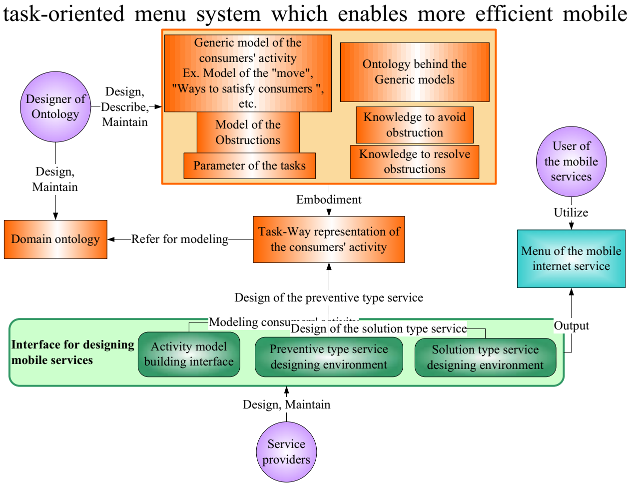


Figure1. Framework for designing task oriented mobile service

service navigation. Result of the experiment shows that task-oriented menu is more efficient for retrieving information [2]. By task, we mean users' problem solving activity in the real world. In the task oriented menu, the users seek for services by the name of the directory which represents a task they are involved in rather than the name of category which might be unfamiliar to them. Users select a menu that is most resemble to what they want to do; "get on the train", "draw cash", for example. It has potential of providing useful information for mobile service users quicker than that of a domain-oriented menu. Value of information depends on the quality of contextual information that contains. By quality, we mean whether the information corresponds to the needs of the users or not. Necessity of information lies in a task, not in a domain. You seek for information when you face a trouble, which is difficult to get over with knowledge at hand, on your way of achieving a task. Such a situation is the context and origin of the necessity for the information.

With backgrounds discussed above, this article proposes a task ontology-based modeling framework for mobile service navigation. Fig.1 depicts the framework of our system where rectangles represent knowledge, rectangles with round corners represent modules and circles represent people. "Service providers" in Fig.1 design users' activity models and mobile internet services through the interface module. Its output is the menu of the mobile internet services that is used by the "User of the mobile services". Although the service providers usually have implicit business models about their own mobile services, they do not have generic task models for representing users' activities. Generic models and task/domain ontologies which are designed by "Designer of Ontology" are referred to by the service providers to obtain concrete models by instantiating the generic models.

"Designer of Ontology", an important role of the authors, designs and maintains ontologies. The authors are specialists for building task ontologies [3], and have experiences of its application to the real world problem solving [5]. Although there are huge numbers of "tasks" in the real world, those have be solved by mobile handset users are small, since they are limited to daily-life tasks done out side home. Furthermore, to organize task concepts is easier than that of domain concepts, because it is independent of domain, is able to be decomposed into subtasks and has a generality in the abstract space. For example, a task concept "buy a ticket for a movie" consists of two task concepts, "buy something" and "receive service (Including model of queuing)". Both concepts can be applied to modeling similar tasks in various domains. Task concept thus has a generality in its nature and hence we can organize its structure at a high level of abstraction.

Demos and Posters of *the 3rd European Semantic Web Conference (ESWC2006)*, Budva, Montenegro, 11th - 14th June, 2006

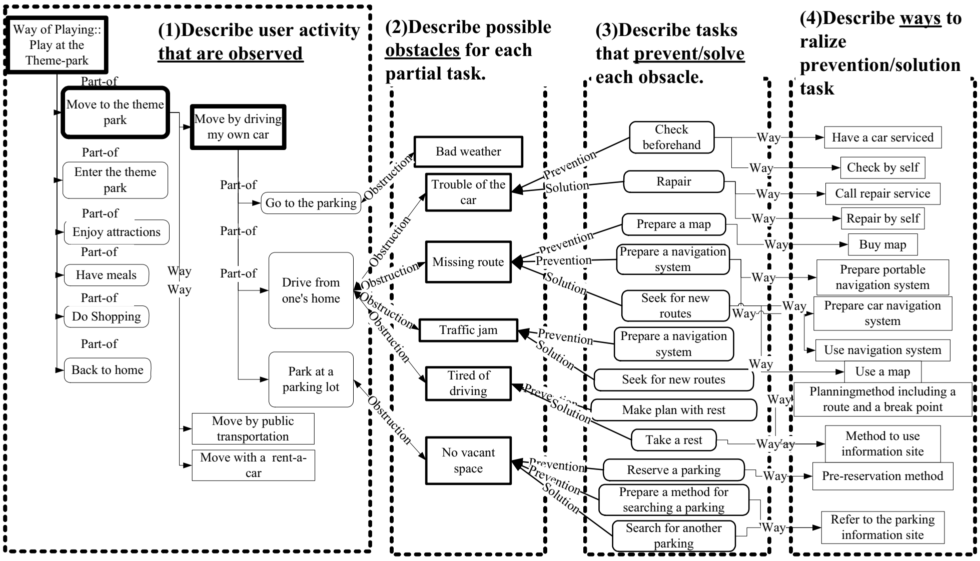


Figure2. An example of modeling process

The authors are investigating reorganization of the mobile services from the viewpoint of task. The approach based on task ontology enables service providers to describe users' activity models in terms of generic task vocabulary which are detached from the domain model. Furthermore, specification of the modeling process based on categorization of users' activity provides them with guidelines. Based on the task ontology, our method contributes to building homogeneous and generic models.

2. Framework

Fig.2 represents a process of building a task-way representation model of users' activity. A dotted rectangle with number (1) corresponds to the basic model of users' activity. It is described by instantiating generic models and/or ontology. Description starts from the task at the level of coarse granularity. Next, ways to achieve the task are linked, and each of the ways is decomposed into a sequence of sub-tasks. Our "way" is similar to "method" of CommonKADS [6] or "how to bundle" of the Business Process Handbook [7] to some extent. Following this process, task of the coarse granularity is decomposed into sub-tasks via a few ways. The area with number (1) represents that a task "Move to a theme park" is achieved by three ways. One of the ways "Move by driving one's own car" is decomposed to three sub-tasks such as "Go to the parking lot", "Drive from one's home" and "Park the car at the parking lot".

An important guideline in this framework is that the model of daily activity is described based on the observation of physical activity on the spot. Cognitive activities such as "plan to move more efficiently" or "learn traffic information beforehand" are not described in the model.

The guideline and modeling process based on decomposition of the task contribute to making modeling process easier and output models more objective. To realize a coherent task-oriented menu structure with a large scale problem, transfer of the modeling technology is important. If we allow modeling non-observable activities, quality of output models varies according to the skill of each model builder. The less knowledge about the task and/or domain he/she has, the worse output models will become. In such a case, process of modeling becomes implicit and we cannot transfer the method to others.

Models of how to prevent/solve problems are described in three steps. Firstly, the designer describes possible obstacles for each partial task. For example, the task "Drive from one's home" has four possible obstacles: "Trouble of the car", "Incomplete route", "Traffic jam" and "Fatigue of driving". Building models of possible troubles is a unique feature compared to previous researches like [2][6]. Since the most valuable mobile service is to solve such problems that occurred, our modeling method copes with obstacles on the spot. Next, the designer describes prevention/solution tasks for each of the obstacles (Fig.2(3)). Lastly, the ways to achieve the tasks are modeled (Fig.2(4)).

Since models of obstacles are described for the task with fine granularity, we can imagine more tasks for their prevention/ solution than conventional modeling methods. In Fig.2, for example, to generate preventive ideas for obstacles about the task "move to theme park" is more difficult than that for the "Park at the parking lot", because the former task is abstract and contains many obstacles according to its interpretation. Based on the decomposition of the task models, our method helps generation of new prevention/solution ideas. With a similar reason, generating ideas about the ways to achieve the prevention/solution tasks is supported by our modeling method.

3. Research Status

The authors are currently building task ontologies and models on the proposed framework. Task model of the activities related to the theme park has been building at the high level of abstraction, and we plan to apply the ontology to other domains (cf. Chap1). At the same time, we are conducting several experiments to evaluate the proposed framework in terms of (1) supports for generation of ideas about mobile services (2) relationship between given ontological information and the quality of the user models output (3) efficiency of the modeling with our guideline, and so on.

4. REFERENCES

[1] http://www.nttdocomo.co.jp/english/corporate/investor_relations/referenc/annual_e.html

[2] T.Naganuma and S. Kurakake., *Task Knowledge Based Retrieval for Services Relevant to Mobile User's Activity,* In Proceedings of the ISWC2005, 959--973, 2005.

[3] Mizoguchi, R., Vanwelkenhuysen, J., and Ikeda, M., *Task Ontology for Reuse of Problem Solving Knowledge,* In Proc. of KB\&KS '95, The Netherlands, 46--59. 1995.

[4] http://www.hozo.jp

[5] Kitamura, Y., Kashiwase, M. Fuse, M., Mizoguchi, R., Deployment of an Ontological Framework of Functional Design Knowledge, Advanced Engineering Informatics, 2004:18(2):115-127 . 2004.

[6] Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W., and Wielinga, B., *Knowledge Engineering and Management- The Common-KADS Methodology* the MIT Press, 2000.

[7] Thomas W. M., Kevin C., and George A.H., *Organizing Business Knowledge-The MIT Process Hand Book*, MIT Press, 2003.

Triple Space Computing: A framework for the new communication paradigm

Reto Krummenacher, Francisco Martin-Recuerda
DERI, University of Innsbruck, Austria
firstname.lastname@deri.org

Martin Murth, Johannes Riemer
Vienna University of Technology, Austria
mm | jr@complang.tuwien.ac.at

Ying Ding, Dieter Fensel
Electronic Web Service GmbH, Innsbruck, Austria

ABSTRACT
Triple Space Computing (TSC) is an emerging communication and coordination paradigm tailored to the Semantic Web and Semantic Web services. In this paper we shortly describe the core ideas behind TSC and the initial efforts of an Austrian research project that aims at realizing the vision of Triple Space Computing.

Keywords
Triple Space Computing, Semantic Web services, service interaction and coordination

1. INTRODUCTION
Semantic Web services promise seamless interoperability of data and applications on a semantic level, thus turning the Web from a world-wide information repository for human consumption only to an infrastructure of distributed computation. Appropriate semantic descriptions of Web services and intelligent mechanisms working upon this, need a solid basement in terms of the underlying semantically enabled communication technologies. Triple Space Computing (TSC) which inherits the publication-based communication model from Tuple Space computing, extending it with semantics, provides solutions in that direction [2]. Instead of sending messages back and forth, applications will communicate by writing and reading RDF triples in a shared persistent information space, the Triple Space.

The TSC project is a research project aiming at developing such a new middleware infrastructure with special support for the Semantic Web and Semantic Web services. TSC is an Austrian national funded project running for 3 years. Currently it is at its early stage. The outcome of the TSC project will be a generic framework and prototype implementation for a Triple Space Computing environment. In this paper we report on the initial ideas of the framework and future plans.

2. TSC CONCEPTUAL MODEL
The TSC framework is based on the evolution and integration of several well-known technologies: Tuple Space computing [3], shared object space [7] and Semantic Web technologies (in particular RDF). It defines the use and extensions of these technologies with regards to their conceptual foundations towards Triple Space Computing.

The first foundation of the TSC framework is Tuple Space computing. The Tuple Space model not only decouples the information exchange in reference, time and space, but also offers a high-level abstraction, namely the communication via the reading and writing of tuples in a space, where a tuple is an ordered set of typed fields. Applying this paradigm offers the advantage of removing the complexity of message-based systems currently used for building Web services. It also offers advantages in terms of reduced development costs, simplicity, extensibility, easy debugging at runtime, and recovery due to persistent storage in the information space.

The second foundation is the application of Web design principles, thus further decreasing the deficiencies of message-based communication and hence improving the scalability of the system. The Web technologies add some additional features that are lacking in current Tuple Spaces: (1) URIs as a unique, well-defined reference mechanism, (2) namespaces as a separation mechanism of information chunks by qualified names, and (3) interlinking of resources by use of foreign URIs as hyperlinks.

The third foundation is the semantics of communication. The semantic descriptions in TSC are based on RDF triples and handled as Named Graphs [1]. Although RDF lacks expressivity for more complex ontology specifications, the triple model [5] provides a simple, but valuable approach for annotating information. Thus, it is considered to be sufficient for prototyping TSC. In more advanced implementations, richer data models than nested triples may be applied — e.g., OWL or rule languages.

Like Tuple Space computing, the basic Triple Space paradigm based on 'persistent publish and read' has a major limitation from the perspective of a client: an application which wants to read a concrete triple or set of triples has to interrupt the main process flow or run a concurrent thread that periodically checks if relevant data is available. Figure 1 illustrates an example of a simple producer-consumer interaction: Process B is the consumer and searches data before it is available in the space. Process A publishes the data. On the left side, process B queries the space (and blocks the main flow) until data is available. On the right side, process

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

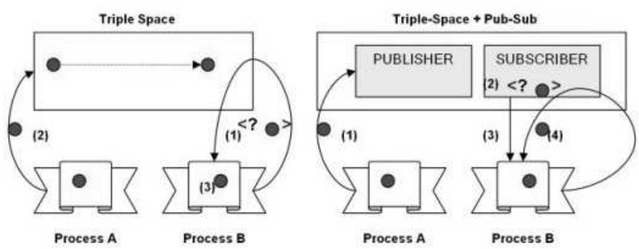


Figure 1: Publish-Subscribe over Triple Spaces.

B is subscribed to the data, and whenever some matching triples are available a notification is sent that indicates that the desired data can be retrieved.

The combination of the publish-subscribe paradigm and TSC defines a more flexible coordination model that extends the idea of a Web-like 'persistent publish and read' communication infrastructure for Web services to an even more decoupled 'persistent publish and subscribe' model.

3. TSC ARCHITECTURE

The envisioned TSC architecture is based on a hybrid architecture that combines P2P and client/server models in a so-called super-peer system [8]. This configuration ends up in a two-tiered system: the upper-tier is composed of a powerful and static server network, while the lower-tier consists of clients that might only be temporarily available and possibly possess limited computational resources (called heavy clients). Embedded devices like smart phones or PDAs, referred to as light clients, will access the Triple Space remotely using the backbone of servers (Figure 2). In that way the TSC framework is not only envisioned to improve service-oriented architectures, but also mobile and even ubiquitous applications [6].

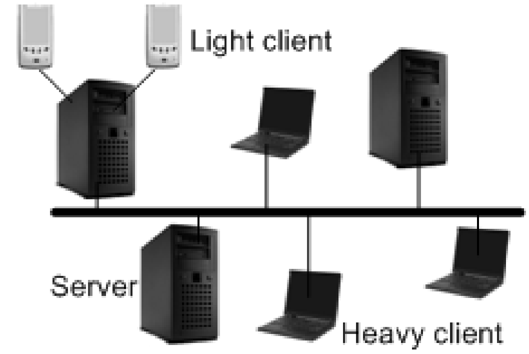


Figure 2: TSC architecture overview.

Heavy clients and servers run a Triple Space kernel (light clients do not) that provides coordination, security and data handling (data mediation, querying and storage) services. The CORSO (Coordinated Shared Objects) framework serves as the starting point to build the TSC kernel. CORSO is a platform for the management of distributed applications in heterogeneous IT environments based on communication via shared objects [7]. The CORSO implementation will thus be extended to support the mediation, replication and communication of semantic data. Yet Another RDF Store (YARS, [4]) is the RDF repository and query engine used by the kernels of the planned prototype to store, manipulate and query the data that each space contains. The idea is however to abstract the coordination service from the underlying data storage infrastructure and to allow the use of arbitrary data stores without altering the core implementation of the TSC kernels.

4. CONCLUSIONS AND FUTURE PLAN

In this paper, we presented initial ideas of a framework for Triple Space Computing, a new communication and coordination infrastructure for the Semantic Web and Semantic Web services. The project responsible for this work still has two years to go. During these two years, we will provide a consolidated TSC architecture and interfaces for the cooperation amongst service components and for the TSC infrastructure as a whole. In particular we will focus on the use of data mediation and query engine components, on data replication, security and privacy mechanisms and investigate how standard service architectures (SOA) can be better applied in TSC. In the end, a running prototype will be provided and the usability will be tested via a case study on how TSC can enhance communication and process coordination of a Semantic Web service execution environment like WSMX (www.wsmx.org).

5. ACKNOWLEDGMENTS

The authors thank the TSC project consortium. The project is funded by the Austrian Federal Ministry under the FIT-IT project grant 809249.

6. REFERENCES

[1] J.J. Carroll, Ch. Bizer, P. Hayes and P. Stickler. *Named Graphs.* Journal of Web Semantics 3(4), 2005.

[2] D. Fensel. *Triple-based Computing.* DERI Research Report: DERI-TR-2004-05-31, May 2004.

[3] D. Gelernter. *Generative Communication in Linda.* ACM Transactions on Prog. Lang. and Systems, 1985.

[4] A. Harth, M. Magni and St. Decker. *Scalable Distributed RDF Storage Infrastructure.* DERI Lion Deliverable 1.02, June 2005.

[5] G. Klyne and J. J. Carroll (eds.). *Resource Description Framework (RDF): Concepts and Abstract Syntax.* W3C Recommendation, February 2004.

[6] R. Krummenacher and Th. Strang. *Ubiquitous Semantic Spaces.* Conf. Supplement of the 7th Int'l Conf. on Ubiquitous Computing, September 2005.

[7] E. Kühn. *Virtual Shared Memory for Distributed Architectures.* Nova Science Publisher, 2001.

[8] B. Yang and H. Garcia-Molina. *Designing a Super-peer Network.* IEEE Int'l Conf. on Data Engineering, 2003.

Towards a Reference Ontology of Functionality for Interoperable Annotation for Engineering Documents

Yoshinobu Kitamura, Naoya Washio, Masanori Ookubo,
Yusuke Koji, Munehiko Sasajima, Sunao Takafuji, Riichiro Mizoguchi
The Institute of Scientific and Industrial Research, Osaka University
8-1, Mihogaoka, Ibaraki, Osaka, Japan, +81-6-6879-8416
{kita, washio,mokubo, koji, msasa, takaj, miz}@ei.sanken.osaka-u.ac.jp

Keywords
Ontology, Knowledge modeling, Knowledge Management

1. INTRODUCTION

Functionality is one of the key concepts in knowledge about artifacts. The goal of this research is to manage the information content of design documents from the viewpoint of functionality of engineering products using semantic annotation about functionality (we call *functional annotation*). It enables engineers to access documents by specifying a function as a metadata query. Such function-oriented management is especially useful in the conceptual design phase to find previous design cases for the same required function and to find related patents. The semantic annotation about function is expected to solve the difficulty of current technical document management based on lexical expressions, that is, many terms (verbs) are used in documents for the same function (and vise versa) without clear semantics.

The research issues for realization of functional annotation are (1) to establish *ontological metadata schema* for consistent functional annotation and (2) to realize *interoperability* among various functional representations. For the former issue, although much research has been conducted on functionality in engineering design (e.g., [2][3][7]), there is neither common definition of function nor enough semantic constraints for consistent functional annotation. For example, "to weld metals" as a manufacturing machine's function in the manner of Value Engineering is not only a function but also implies a certain way to achieve the goal, say, "the metals are fused". This issue, that is, distinguishing "what to achieve" from "how to achieve", is not a terminological but ontological. Although PhysSys [1] is a well-established ontology in engineering domain, it does not include functionality.

On the latter issue on interoperability, firstly, there are some taxonomies of verbs for generic functions such as the *generally valid functions* [7] and the (reconciled) *functional basis* in the NIST Design Repository Project [3]. Secondly, many functions are captured for the same use of the same product according to the scope of interest. For example, a function of an electric fan can be captured as "to move air", "to cool human body" or "to make human comfortable". These differences are also not terminological but ontological, because such functions are based on different conceptualizations. A functional annotation schema proposed in [6] uses the functional basis [3] as taxonomy with neither ontological consideration nor interoperability.

The authors have investigated functionality of devices for long years and established an ontology-based framework for functional models [4]. It includes a device-centered functional ontology [5] and a functional concept ontology as functional taxonomy. It has been deployed successfully in industry [4].

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

On the basis of the previous effort, the authors propose a multi-layered framework of ontology-based semantic annotation about functionality (called *Funnotation* (abbreviation of FUNctional anNOTATION) hereafter). It includes a metadata schema in OWL based on our functional ontologies [4][5]. Metadata in RDF based on the schema shows the function of the artifact mentioned in the document. Then, a document search system using the functional metadata helps engineers access to web documents about design in terms of what they want to realize, i.e., function, independently of lexical terms in the documents.

For interoperability of functional metadata, *Funnotation* includes a reference ontology of function (called *FuRO*) which defines categories (classes) of various kinds of function. It aims at clarifying ontological difference between the functional taxonomies and at enabling translation between them. It is an extension of coverage of our functional ontology to cover broader sense of function. A part of *FuRO* has been shown in [5]. This paper discusses its role in interoperability of functional metadata.

2. FUNCTIONAL ANNOTATION

Figure 1 shows an overview of the *Funnotation* framework for functional annotation. Its main schema consists of F-Core schema and F-Vocab schema. F-Core schema defines fundamental classes such as *entity, device, stuff, energy, function* and *way (of function achievement)* together with properties among them such as *has-function* and *selected-way.* The *way* (of function achievement) represents background knowledge of the function achievement (function decomposition [7]) where a function is achieved by a series of finer-grained (part) functions. The *has-function* property is a relation between an (subclass of) *entity* and a *function* where the entity can perform the function as an agent. The *selected-way* property is a relation between a *function* and a *way* where the function is achieved using the way of function achievement in a device. The F-Vocab schema defines a hierarchy of generic functions based on the functional concept ontology [4].

The *Funnotation* schema implemented in OWL enables us to describe metadata in RDF representing functionality of engineering devices in documents. For example, a part of a metadata m_a in Fig. 1 shows that the device appearing in annotated document d_a (a filter) can perform an instance of the *separating* function class defined in the schema. This metadata is annotated to the term "extract" in d_a. The metadata m_b shows that the distiller (the device mentioned in the document d_b) has the same *separating* function. It is, however, annotated to the different term "refine" in d_b. In this manner, functional metadata shows device's functions independently of the terms in documents and indicates pointers (URLs) to the original documents and/or terms. Moreover, the metadata shows how to achieve a function, i.e., in this case, two different ways (the filtering way and the distilling way) to achieve the same separating function.

By querying such functional metadata, a semantic search system provides access to the annotated documents based on the hierarchy of functions and/or relationship between functions and ways. Using the example in Fig. 1, if an engineer specifies the "separate" function as a goal function in a query, the system provides him/her with hyperlinks to the both documents d_a and d_b. We have implemented the search system using Jena and SPARQL.

3. REFERENCE ONTOLOGY OF FUNCTION

The *Funnotation* framework realizes interoperability of functional metadata based on a *reference ontology of function* (*FuRO*) as shown in Fig. 1. It defines generic (upper) classes of various kinds of function. By reference ontology, we here mean an ontology referred to for categorizing existing definitions of function and mapping them (in comparison with "reference for system design" such as the ISO's OSI network reference model). For example, a *device function* implies changes of entities (behaviors) within the system boundary, while an *environmental function* includes changes outside of the system boundary, especially, those related to users. For instance, an electric fan performs moving-air function as a *device function* and cooling function for human body as an *environmental function*, where the cool-down effect by wind is on human body and thus out-side of the system boundary. This cooling function implies physical changes (called *physical environmental function*), while an *interpretational* function sets up one of the necessary conditions of human's cognitive interpretation. The examples of the latter kind are "to make a man comfortable" function of the electric fan and "to inform time" function of a clock. In the literature, there are similar types of function such as "environment function" [2].

Moreover, we recognize the some kinds of *quasi-functions*. Although the authors do not consider them as kinds of function, it is found that a quasi-function is confused with a function. For example, a *function-with-way-of-achievement* implies a specific way of function achievement as well as a function. Its examples include washing, shearing, adhering (e.g., glue adheres A to B) as well as welding mentioned in Introduction. Because meaning of this type of function is impure, we regard this quasi-function.

Each function in the taxonomies is classified into a class of function in *FuRO*. Our functional concept ontology (F-Vocab schema) defines functions strictly from the device-centered viewpoint in three major categories of functions; base-functions, meta-functions and function types [4][5]. All base-functions are categorized into *flowing-object function* in *FuRO*. It represents that a device as a black-box changes a value of physical quantity of objects (or stuff) flowing through the device.

On the other hand, as an example of other taxonomy of function, the functions defined in the functional basis [3] (FB hereafter) are categorized into different classes in *FuRO*, though many of them are classified to the *flowing-object function*. For example, the "indicate" function in FB is categorized as an *interpretational function* in *FuRO* which requires human's cognitive interpretation. The "link" function in FB is a *function-with-way-of-achievement,* because it is defined as "to couple flows (objects) together by means of an intermediary flow" [3].

The mapping between F-Vocab and FB can be done via *FuRO*. Such functions categorized into the same class in *FuRO* can be associated with each other directly. In the simplest case, there is one by one mapping such as "couple" in FB and "combine" in F-Vocab. There are, however, many mismatches due to difference of categorization. On the other hand, if functions

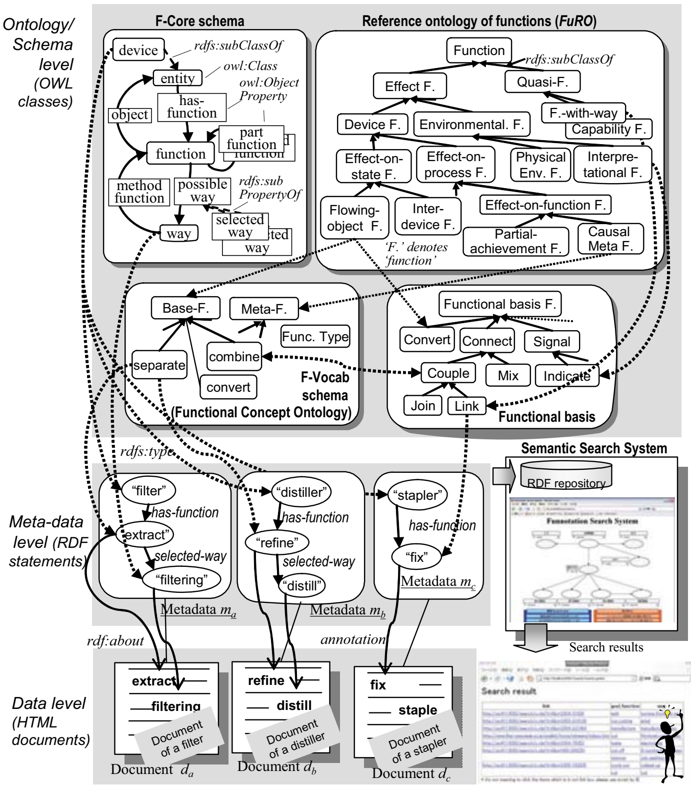


Figure 1. Overview of *Funnotation*: A Framework for Semantic Annotation about Functionality for Engineering Documents

are categorized into the different classes in *FuRO*, the mapping become complex. For example, "link" in FB is translated into "combine" in F-Vocab plus the "intermediate-object" way for combining. Thanks to *FuRO*, such ontological difference becomes explicit and thus we can realize the mapping without loss of information of impure terms in FB.

4. CONCLUSION

The reference ontology of functions can be used to clarify ontological differences between the functional taxonomies and to enable translation of functional metadata between them.

REFERENCES

[1] Borst, P. Akkermans, H., and Top, J, "Engineering Ontologies", Human-Computer Studies, 46(2/3), 365-406, 1997.

[2] Chandrasekaran, B, Josephson, J.R., 2000, "Function in Device Representation", Engineering with Computers, 16(3/4), 162-177.

[3] Hirtz, J., Stone, R.B., McAdams, D.A., Szykman, S., Wood, K.L., 2002, A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts, Research in Eng. Design, 13, 65-82.

[4] Kitamura, Y., Mizoguchi, R., Ontology-based Functional-Knowledge Modeling Methodology and its Deployment, Proc. of EKAW 2004, 99-115, 2004.

[5] Kitamura, Y., Koji, Y., Mizoguchi, R., An Ontological Model of Device Function and its Deployment for Engineering Knowledge Sharing, The Formal Ontologies Meet Industry Workshop, 2005.

[6] Kopena, J.B., Regli, W.C., Functional Modeling of Engineering Designs for the Semantic We", IEEE Data Engineering Bulletin, 26(4), 55-62, 2003.

[7] Pahl, G., and Beitz, W., 1998, Engineering Design - a Systematic Approach. The Design Council.

OWL Linker: A Tool for E-Connecting OWL Ontologies

Farnaz Derakhshan
Department of Computer Science
The University of Liverpool
L69 3BX, UK
farnaz@csc.liv.ac.uk

ABSTRACT

Developing ontologies in many different domain areas has led to a huge amount of distributed information. On one hand, correlating this distributed information seems so worthwhile for providing additional information and reusing ontologies. On the other hand, keeping ontologies in small modules would be helpful for maintaining and managing ontologies. Therefore, recently composing and decomposing ontologies become of the most significant topics in the semantic web researches and the related applications. In this poster, a new tool called "OWL Linker" which provides an environment for combining OWL ontologies would be presented. This novel tool is a plug-in extension of Protégé which provides a very flexible user interface for connecting two separate ontologies based on the e-connection method such that user is able to browse, edit and manipulate link properties and also define individuals and restrictions for these new types of properties. Providing reasoning and updating e-connected ontologies are the other main facilities of this system.

Keywords

OWL Linker, OWL Ontology, E-Connection, Protégé, ontology connection, Protégé plug-in.

1. INTRODUCTION

Developing ontologies in many different domain areas has led to a huge amount of distributed information. Correlating this distributed information is very valuable for providing additional information and reusing ontologies. In addition, keeping ontologies in small modules and using the appropriate connection (when it is necessary) will grow the performance of maintaining and managing ontologies. Therefore, recently composing and decomposing ontologies become of the most significant topics in the semantic web researches and the related applications. This poster presents a new tool called "OWL Linker" which provides an environment for combining OWL ontologies. In fact using this tool user can connect two separate OWL ontologies with e-connection method.

E-connection is an approach for connecting different types of knowledge representation systems that in this case it is used for OWL-DL ontologies. In the next section, this method will be explained.

Technically, OWL Linker is a plug-in extension of Protégé [1, 2] which is a successful ontology editor with a large community of users. It supports OWL ontologies in its OWL Plug-in. The well-designed graphical user interface, expandability, Plug-in-based architecture, and connecting to reasoners (such as RACER) are of the main characteristics of Protégé such that these features perform an appropriate infrastructure for this tool. Next, after a short explaining of the theory of the e-connection, OWL Linker and its features will be described.

2. E-Connecting OWL Ontologies

Among several methods for combining logics, E-connection is a new method which is robust in computational behavior and is defined for abstract description systems [3-5]. Since design of OWL ontologies has been influenced by Description Logics [6], thus, in this work for composing OWL ontologies the method of e-connecting of Description Logic systems has been applied[1].

In this part, we skip the mathematical aspects and describe the general idea of E-connections of description logics systems with an example.

As an example, suppose Medical and People ontologies are completely distinct and they have no common name in their classes or properties. The desired links for connecting these ontologies are:

$$\varepsilon = \{E1 = \text{hasDisease}, E2 = \text{prescribe}, E3 = \text{diagnose}\}$$

And the Description of People and Medical ontology are as follows (Figure 1).

$\text{person} \equiv = 2\ \text{hasParent.person}$
$\text{parent} \equiv \exists\ \text{hasChild.person}$
$(\text{hasChild})^{-1} \equiv \text{hasParent}$
$\text{hasParent} \sqsubseteq \text{hasAncestor}$
(Mari,Ali): hasChild
(Ali,Ebi): hasParent
(Ebi, David): hasAncestor

$\text{SurgeryOperation} \sqsubseteq \text{treatment}$
$\text{pills} \sqsubseteq \text{treatment}$
$\text{diabetes} \sqsubseteq \text{Disease}$
$\text{hepatitis} \sqsubseteq \text{Disease}$
diabetesA: diabetes
diabetesB: diabetes
(diabetesA, pillsA): hasRemedy
(diabetesB,insulin): hasRemedy

Figure 1- People Ontology & Medical Ontology

Now it is possible to define a new class which uses the link property between ontologies. For example, diabetesSuspicious is a person who has at least one ancestor with at least diabetes disease as shown in Figure 2.

$$\text{diabetesSuspicious} \sqsubseteq \text{person} \sqcap (\exists\ \text{hasAncestor}(\exists\ \text{hasDisease.diabetes}))$$
(David,diabetesB): hasDisease

Figure 2- An example of link relation

In this way, we can keep both ontologies separate and just by defining e-connection get the facility of combining them, defining new classes

[1] There are also other methods for connecting ontologies such as C-OWL. While E-Connection link ontologies using binary relation (property), C-OWL use subsumption relation. More information is available in:
[1] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, H.Stuckenschmidt: Contextualizing Ontologies. Journal of Web Semantics 2004.

and new constraints. With respect to the above assertions, Ali: diabetesSuspicious is satisfiable. Because from the first ontology, David is one of Ebi's ancestor, and Ebi is Ali's parent. And because hasParent is subsumed by hasAncestor, David is Ali's ancestor as well. On the other hand, based on the Medical ontology, diabetesB is a kind of Diabetes. So if David has diabetesB, Ali would be diabetesSuspicious by the definition of the above link property axiom.

3. OWL Linker

OWL Linker has been developed with Java and it benefits from using Protégé APIs. The main characteristics of Protégé such as well-designed graphical user interface, expandability, plug-in-based architecture, and connecting to reasoners perform an appropriate infrastructure for developing this tool. So OWL Linker is a new tab in the Protégé environment.

OWL Linker provides novel facilities for e-connecting distinct ontologies. It has a friendly user interface that user is able to browse and select the desirable two OWL ontologies. Next s/he can create, edit and manipulate link properties between these ontologies. Assigning new individuals to the classes and link properties is possible as well. In addition, defining new classes is possible to define new restriction using link properties which is too similar to define entities and axioms in Protégé.

One of the other main features of this tool is that if user leaves the session, then next time s/he wants to do any updates and changes on e-connection of ontologies, there exists the capability of reloading the previous environment for changing or updating of the link properties, classes and other entities of two ontologies. It means system will save all the activities during the session.

The next characteristic of the OWL Linker is its capability of connecting to reasoners to perform reasoning services over the e-connected ontologies. Currently the well-known reasoners such as RACER [7] have not been designed for performing such reasoning. Therefore to obtain the reasoning facility for e-connected ontologies, these two distinct systems and their e-connection links has to be combined as an integrated system. For this reason, OWL Linker provides such combined ontology which the definition of the link properties is similar to the standard OWL syntax of properties, however, the domain and range of properties are from two separate ontologies and also to differentiate from the usual properties system automatically adds an appropriate prefix to the name of the link properties. Creating the combined ontology provides an appropriate input for reasoners. In this way, reasoners can do reasoning just like the other Description Logics systems.

For instance, RACER as a powerful Description Logics reasoner system can be accessed from OWL Linker. By pressing the reasoning buttons in the main panel of the OWL Linker, connecting to RACER will be provided and the result of reasoning services of RACER will be displayed into OWL Linker environment.

4. Related Work

There exists a similar and progressive work in MindSwap group in The University of Maryland which results two new environments: SWOOP [8, 9] semantic web editor and Pellet [10] reasoner. SWOOP enables user to render different types of ontologies, edit and visualize classes, properties, individuals and define logical class characteristics as OWL expressions. Ontology linking with E-connection method [11] is of the other features of SWOOP. To perform e-connection between ontologies, it adds some extension to the normative OWL abstract syntax for link properties [11].

Currently, for reasoning services, SWOOP just can access to Pellet as the default reasoner by selecting Pellet in Reasoning Combo in the editor environment. Pellet [11, 12] is a novel tableau based DL reasoner specifically developed for working with OWL ontologies and also supports multi-ontology reasoning using E-Connections[13]. In comparison the advantage of our work is that the other existing DL reasoners like RACER also can be used to perform reasoning on e-connected system.

5. SUMMARY

To sum up, since from the implementation point of view requiring suitable tools and applications to support theoretical aspects are inevitable, this poster will present a tool to provide connecting ontologies with e-connection. In addition, this tool has the facility of connecting to the existing reasoners like RACER for reasoning over the linked ontologies can be provided.

6. ACKNOWLEDGMENTS

Finally I would like to thank Professor Frank Wolter and Dr. Clare Dixon for their valuable help and direction especially on the E-connection theory.

7. REFERENCES

[1] H. Knublauch, R. Fergerson, N. F. Noy, and M. A. Musen, "The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications," Third International Semantic Web Conference, Hiroshima, Japan 2004.

[2] H. Knublauch and M. A. Musen, "Editing Description Logic Ontologies with the Protege OWL Plugin," International Workshop on Description Logics, Whistler, BC, Canada 2004.

[3] C. Lutz. O. Kutz , F. Wolter and M. Zakharyaschev, "E-connections of abstract description systems," *Artificial Intelligence*, vol. 156, 1-73, 2004.

[4] O. Kutz, F. Wolter, and M. Zakharyaschev, "Connecting abstract description systems," Proceeding of KR 2002, 215-226, 2002.

[5] B. Parsia. B. C. Grau, E. Sirin, "Tableau algorithms for E-connections of Description Logics," 2004.

[6] I. Horrocks, P. F. Patel-Schneider, and F. v. Harmelen, "From SHIQ and RDF to OWL: The making of a Web Ontology Language," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 1, 7-26, 2003.

[7] V. Haarslev and R. Moller, "Description of the RACER System and its Applications," International Workshop on Description Logics, Stanford, USA 2001.

[8] A. Kalyanpur, E. Sirin, B. Parsia, and J. Hendler, "Hypermedia inspired ontology engineering environment: Swoop," 3rd International Semantic Web Conference, Hiroshima, Japan 2004.

[9] A. Kalyanpur, B. Parsia, and J. Hendler, "A Tool for Working with Web Ontologies," Journal on Semantic Web and Information Systems,1, 2005.

[10] B. Parsia. E.Sirin, "Pellet:An owl-dl reasoner," 2004.

[11] B. C. Grau, B. Parsia, and E. Sirin, "Working with Multiple Ontologies on the Semantic Web," UMIACS, 2004.

[12] E. Sirin and B. Parsia, "Pellet:An owl-dl reasoner," ISWC2004, Hiroshima, Japan 2004.

[13] B. C. Grau, B. Parsia, and E. Sirin, "Combining OWL Ontologies using e-connection," UMIACS January 2005.

From Personal Ontologies to Socialized Semantic Space

Jason J. Jung
INRIA Rhône-Alpes
ZIRST 655 avenue de l'Europe, Montbonnot
38334 Saint Ismier cedex, France
j2jung@intelligent.pe.kr

Jérôme Euzenat
INRIA Rhône-Alpes
ZIRST 655 avenue de l'Europe, Montbonnot
38334 Saint Ismier cedex, France
Jerome.Euzenat@inrialpes.fr

ABSTRACT

We have designed a three-layered model which involves the networks between people, the ontologies they use, and the concepts occurring in these ontologies. We propose how relationships in one network can be extracted from relationships in another one based on analysis techniques relying on this network specificity. For instance, similarity in the ontology layer can be extracted from a similarity measure on the concept layer.

Keywords

Semantic web, social network

1. INTRODUCTION

Social network is built from the explicit assertion by users that they have some relation with others or by the implicit evidence of such relations (e.g., co-authoring). In order to support efficient collaboration between users, we propose a three-layered architecture that is capable of capturing the semantics emerged from communities. These semantics are discovered from analyzing the user's social activities on the semantic space. In order words, while building the personal ontologies, the social activities such as linking to a certain user and referring to a domain ontology can represent the corresponding user's semantic preferences. In contrast, as a related work, a tripartite model (actors, concepts, and instances) in [2] has focused on the personal activities based on tagging instances. Also, the similarity measurement for socializing the users is done by co-occurrence analysis with instances and concepts applied by them.

2. THREE-LAYERED ARCHITECTURE

We have designed the three-layered architecture composed of social, ontology, and concept layer. In **social layer** ($\mathcal{S}$), nodes are representing people, and relations are the connections between peoples. It is a directed graph $\langle N_S, E_S^{knows}\rangle$, where N_S is a set of person and $E_S^{knows} \subseteq N_S \times N_S$ the set of relations between these persons.

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

Ontology layer $\mathcal{O}$ is a network $\langle N_O, E_O^i\rangle$, in which N_O is a set of ontologies and $E_O^i \subseteq N_O \times N_O$ the relationships between these ontologies. Two main kind of relations are i) *import* when some ontology explicitly import another ontology, and ii) *refer* when some ontology uses some concept defined in another ontology. The objective relationship from the $\mathcal{S}$ to the $\mathcal{O}$ is through the explicit usage of ontology by a user which can be expressed by a relation: $Use \subseteq N_S \times N_O$.

In **concept layer** ($\mathcal{C}$), nodes are concepts, and links are the numerous kinds of links that can be found in ontologies. It is a network $\langle N_C, E_C^i\rangle$, in which N_C is a set of entity of an ontology (classes, properties, individuals) and $E_C^i \subseteq N_C \times N_C$ the relationships between these entities. This time the relationships are far more numerous and depends on the kind of entity considered. Such relationships are i) *subClass* linking a class to its subclasses; ii) *superClass*(=subClass^{-1}) linking a class to its super classes; iii) *sibling* linking a class to its siblings; iv) *disjoint* linking a class to the classes it is explicitly disjoint with; v) *property*(=domain^{-1}) linking a class to its properties; vi) *range*$^{-1}$ linking a class to the properties that refer to it. The objective relationship from the $\mathcal{O}$ to the $\mathcal{C}$ is through the definition of concept in an ontology which can be expressed by a relation: $Defines \subseteq N_O \times N_C$. However, this notion of definition is not easy to catch: it could be based on either the assertion of a constraint on some ontology entity or the namespace in which entity belongs. Wewill consider the namespace in the following.

3. INFERRING RELATIONSHIPS

This three-level semantic social network does not bring in itself new improvement for our peer-to-peer sharing application. In order to provide new insight in the possible collaborations it is necessary to analyze these networks and to propagate information from one layer to another. It is assumed that user behaviors is semantically socialized. We explain how, starting from the lower concept layer, it is possible to enrich the upper ontology layer and social layers with new relations from which social network analysis helps finding relevant peers. Besides the numerous relationships that can be found by construction of the concept layer, new relationships can be inferred between the entities. One particular relationship that will be interesting here is *similarity*. In order, to find relationship between concepts from different ontologies, identifying the entities denoting the same concept is a very important feature. As a matter of fact, most of the matching algorithms use some similarity measure or distance in order to match entities. In the spirit of network analysis, they can be defined from the structure of the network. For instance, [1], defines all possible similarities (e.g., Sim_C, Sim_R, Sim_A) between classes, relationships, attributes, and instances. Given a pair of classes c and c' from two different ontologies, $Sim_C \in [0,1]$ is defined as

$$Sim_C(c,c') = \sum_{E \in \mathcal{N}(C)} \pi_E^C MSim_Y(E(c), E(c')) \quad (1)$$

where $\mathcal{N}(C) \subseteq \{E^1 \dots E^n\}$ is the set of all possible relationships in which classes participate, e.g., subclass, instances, or attributes. The weights π_E^C are normalized (i.e., $\sum_{E \in \mathcal{N}(C)} \pi_E^C = 1$). Thus, if we consider class labels (L) and three relationships in $\mathcal{N}(C)$, which are the superclass (E^{sup}), the subclass (E^{sub}) and the sibling class (E^{sib}), Equ. 1 is rewritten as:

$$\begin{aligned} Sim_C(c,c') &= \pi_L^C sim_L(L(A_i), LF(B_j)) \\ &+ \pi_{sup}^C MSim_C(E^{sup}(c), E^{sup}(c')) \\ &+ \pi_{sub}^C MSim_C(E^{sub}(c), E^{sub}(c')) \\ &+ \pi_{sib}^C MSim_C(E^{sib}(c), E^{sib}(c')). \quad (2) \end{aligned}$$

where the set functions $MSim_C$ compute the similarity of two entity collections. A distance between two set of classes can be established by finding a maximal matching maximising the summed similarity between the classes:

$$MSim_C(S,S') = \frac{\max(\sum_{\langle c,c'\rangle \in P(S,S')} (Sim_C(c,c'))}{\max(|S|,|S'|)}, \quad (3)$$

in which P provides a matching of the two set of classes. Methods like the Hungarian method allow to find directly the pairing which maximises similarity. The OLA algorithm is an iterative algorithm that compute this similarity [1]. This measure is normalised because if Sim_C is normalised, the divisor is always greater or equal to the dividend.

A normalized similarity measure can be turned into a distance measure by taking its complement to 1 ($E_C^{dist}(x,y) = 1 - Sim_C(x,y)$). Such a distance introduces a new relation E_C^{dist} in the concept network $\mathcal{C}$. This relation in fact defines a distance network as introduced above.

Then, it can be used for computing a new distance at the ontology level. Again, a distance between two ontologies can be established by finding a maximal matching maximising similarity between the elements of this ontology and computing a global measure which can be further normalised. Thus, **ontology distance** can be computed. Given a set of ontologies N_O, a set of entities N_C provided with a distance function $E_C^{dist} : N_C \times N_C \longrightarrow [0\ 1]$ and a relation $D : N_O \times N_C$, the distance function $E_O^{dist} : N_O \times N_O \longrightarrow [0\ 1]$ is defined as

$$E_O^{dist}(o,o') = \frac{max(\sum_{\langle c,c'\rangle \in P(D(o),D(o')} E_C^{dist}(c,c'))}{max(|D(o)|, |D(o')|} \quad (4)$$

which is the measure that is used in the OLA algorithm for deciding which alignment is available between two ontologies [1]. However, other distances can be used such as the well known single, average and multiple linkage distances.

This ontology distance introduces a new relation on the ontology layer. This measure provides a good idea of the distances between ontologies. These distances, in turn, are a clue of the difficulty to find an alignment between ontologies. It can be used for choosing to match the closest ontologies with regard to this distance. This can help a newcomer in a community to choose the best contact point: the one with who ease of understanding will be maximised.

Once these measure on ontologies are obtained, this distance can be further used on the social layer. As we proposed it is possible to think that people using the same ontologies should be close to each other. It is possible to measure the **affinity** between people from the similarity between the ontology they use. Given a set of people N_S, a set of ontologies N_O provided with a distance $E_O^{dist} : N_O \times N_O \longrightarrow [0\ 1]$ and a relation $Uses : N_S \times N_O$, the affinity is the similarity measure defined as

$$E^{aff}(p,p') = \frac{1 - max\left(\sum_{\langle o,o'\rangle \in P(Use(p),Use(p'))} 1 - E_O^{dist}(o,o')\right)}{max(|Use(p)|, |Use(p')|)}$$

Since this measure is normalised, it can be again converted to a distance measure through complementation to 1. Introducing the distance corresponding to affinity in the social network allows to compute the affinity relationships between people with regard to their knowledge structure.

4. CONCLUSION AND FUTURE WORK

In order to improving the collaborative sharing and exploitation of this knowledge, we have proposed a three-layered architecture for constructing socialized semantic space from personal ontologies. This space not only supports the relations within a layer, but also the propagation of relations between layers. We have provided the principles for extracting similarity between concepts and propagating this similarity to a distance and an alignment relation between ontologies. This distance relation can be used for discovering affinity in the social network.

There remains important issues to be investigated: all these networks are not equal and their exploitation with classical social network analysis tools can be meaningless (in the same sense that considering the “loves” and “hates” relations as the same would lead to problems). It is thus important to characterise the various relations that were provided with regard to the measures that can be used on them.

5. REFERENCES

[1] J. Euzenat and P. Valtchev. Similarity-based ontology alignment in OWL-Lite. In R. L. de Mántaras and L. Saitta, editors, *Proc. of 16th ECAI*, pages 333–337. IOS Press, 2004.

[2] P. Mika. Ontologies are us: A unified model of social networks and semantics. In *Proc. of 4th ISWC*, pages 522–536, 2005.

TagFS: Bringing Semantic Metadata to the Filesystem

Simon Schenk, Olaf Görlitz, Steffen Staab
Institute for Computer Science
University of Koblenz
http://isweb.uni-koblenz.de
{sschenk, goerlitz, staab}@uni-koblenz.de

1. INTRODUCTION

Tagging has recently become very popular because of internet applications like del.icio.us and flickr which allow easy categorisation of personal information plus sharing it with a large community. These tools are centralised internet services enabling users to collaborate, organise and share personal information. Most tagging applications are tailored to a specific set of information objects that the user manages online at a centralised storage site. To push tagging towards becoming a significant part of user's everyday work it should be integrated in a broad range of desktop applications. Today the tool most commonly used for structuring knowledge among average users is the filesystem. In the following we introduce TagFS which allows tagging of files as well as tag-based browsing for arbitrary information objects on top of the local filesystem. Tagging information is stored in RDF in order to enable easy integration with semantic web and semantic desktop applications.

As a use case, attending a conference is a scenario in which many information objects become relevant: photos taken at the conference, electronic tickets and reservations, electronic papers, etc. However, when surveying latest photo shots for sharing on a photo server, when compiling the latest travel cost statements, or when sorting the papers to be read by colleagues, hierarchical organisation of information objects is inconvenient. In contrast, tags allow for structuring an information object into the different dimensions for which it is relevant.

Keywords

Filesystem management, semantic desktop, tagging

2. ARCHITECTURE

Representing information about tagging in an ontology has the advantage that extensions of the data model and integration with other semantic aware applications are easy to realise. Figure 1 depicts the ontology used for TagFS. TagFS[1] provides filesystem operations (list directory, create directory, create file, delete file, etc.) that let legacy applications work seamlessly with TagFS while new applications can utilise the full power of the tagging-based infrastructure through extended interfaces on top of the metadata store. Though TagFS provides all the usual filesystem operations, the semantics of these operations have been changed significantly.

2.1 Metadata and Views

TagFS manages all filesystem information as metadata in a RDF repository following our tagging ontology. All actual files are stored in a file repository, currently an underlying conventional filesystem, using unique IDs internal to TagFS. The metadata-based approach allows for large flexibility. In particular, it allows to treat other information objects, such as bookmarks, addresses or emails, equally like files.

The semantics of filesystem operations are defined by queries and update operations on the metadata, i.e. the RDF graph, plus some minor bookkeeping for physical storage. We also define *views* that translate into SPARQL queries. A view, i.e. the corresponding SPARQL query, is applied on a RDF graph and always results in another RDF graph allowing for functional composition.

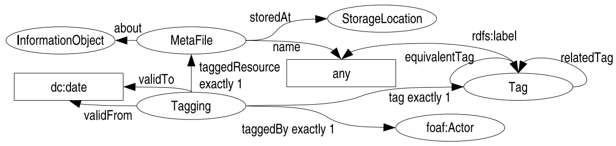


Figure 1: The Tagging Ontology

2.2 Working Directory vs. Context

An important view, called 'hasTag' was defined to select files related to a given tag. For example, the view hasTag(/, 'paper') returns from the complete metadata repository (denoted by '/') all MetaFile identifiers tagged by 'paper' and their associated data. Similarly, hasTag(hasTag(/, 'paper'), 'WWW2006') composes two views returning all MetaFile identifiers tagged 'paper' and 'WWW2006'.

We provide a shorthand query notation for the 'hasTag' view and its composition, e.g. '/paper/WWW2006' for the running example, being equivalent to '/WWW2006/paper', because the composition of hasTag-views is commutative.

[1] This research was partially supported by the European Commission under contract FP6-027026, Knowledge Space of semantic inference for automatic annotation and retrieval of multimedia content - K-Space. The expressed content is the view of the authors but not necessarily the view of the K-Space project.

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006
.

Obviously, the shorthand syntax for hasTag-views shows many correspondences with common directory names. When a legacy application changes a current working directory, e.g. from '/' to '/a/b/c', the semantics of subsequent filesystem operations ('ls', 'rm *', etc.) will be defined as being executed on the RDF graph returned by the query '/a/b/c' rather than on the complete repository denoted by '/'. We will call the result of a query like '/a/b/c' which is acting as a kind of "working directory" a *working context* or simply *context* and the query itself the *context description*.

Hence, the specification of a directory like '/a/b/c' becomes a complex metadata graph instead of a simple node (i.e. directory) in a tree with subnodes (i.e. subdirectories). To simulate the listing of directories (ls), we provide another view, called LS, with signature LS: graph → list returning a list of file- and subdirectory names. This is neccessary to map view contents to the flat representation required by the common filesystem interface.

2.3 Modifying Filesystem Information

The set of possible *modification operations* on the metadata repository is described in Table 1 with *ctxt* being a context and *metafile* being the involved MetaFile identifier.

`addProperties (ctxt, metafile)`	Add statements to the metadata graph so that metafile is included in context ctxt.
`addProperties (ctxt1, ctxt2)`	For all MetaFiles `metafile` in context ctxt2 `addProperties(ctxt1, metafile)`.
`removeProperties (ctxt, metafile)`	Remove statements from the metadata graph so that metafile is no longer in context ctxt.
`removeProperties (ctxt)`	For all MetaFiles `metafile` included in context ctxt `removeProperties(ctxt, metafile)`.

Table 1: Repository Operations

2.4 Mapping Filesystem Operations to TagFS

Table 2 summarizes the mapping of filesystem operations to repository operations. Context descriptions have the form of filesystem paths. If no context description is explicitly passed as a parameter, we assume that it is implicitly given by the path to the working directory. When calling an operation for modifying metadata, contexts are resolved from context descriptions by executing the corresponding query.

The reader may note that only operations like *read, write, create, copy* need to distinguish whether the referenced object was a proper file or rather another kind of information object, such as a bookmark or address. To achieve this distinction, these operations are delegated to *ClassHandlers*, which implement them specifically for a certain class of information objects. For local files, the operations are then forwarded to the underlying storage system (in our case the underlying legacy file system).

3. IMPLEMENTATION

Our Linux-based implementation uses fuse[2] and fuse-j[3] which provide access to the Linux filesystem API from userspace and expose corresponding java-bindings. Sesame 2.0[4] is used as RDF repository. Views are not implemented as simple queries but as objects with a method taking a graph and additional parameters and returning a graph. The view object also provides a method which, given a graph, view

`move oldCtxdesc/File newCtxdesc/File`	removeProperties(oldCtxdesc, MetaFile); addProperties(newCtxdesc, MetaFile)
`rename File newFile`	removeProperties(ctxdesc, MetaFile); create new metafile with new file name; addProperties(ctxdesc, newMetaFile)
`delete File`	removeProperties(ctxdesc, MetaFile)
`create subdirectory`	addProperties(subdirectory, placeholder[1])
`rename oldCtxdesc newCtxdesc`	addProperties(newCtxdesc, oldCtxdesc); removeProperties(oldCtxdesc); make sure not to remove statements in the intersection of old and new context.
`delete ctxdesc`	removeProperties(ctxdesc)
`link File newCtxdesc`	addProperties(newContext, MetaFile)
`link File newCtxdesc/newFile`	Create new meta file referencing the same information object as old MetaFile; addProperties(newCtxdesc, newMetaFile).
`link ctxdesc1 ctxdesc2`	addProperties(ctxdesc2, ctxdesc1)
`create File`	Create a new information object referenced by MetaFile, addProperties(ctxdesc, MetaFile)
`read File`	Read from referenced information object
`write File`	Write to referenced information object
`copy File`	like create file followed by a write

Table 2: Mapping Filesystem Operations to TagFS

parameters and a file returns a graph containing all statements which are neccessary for the File to appear in the context described by the parameters.

The view hasTag returns a subgraph containing all `MetaFiles` which have a `Tagging` relation to the tag with the label given as parameter, except a tagging has a validTo property which refers to a time in the past. Additionally, all attributes and `Tagging` relations of the matched `MetaFiles` are included in the result graph. `addProperty` returns a graph containing taggings, which also include taggedBy and validFrom properties. `removeProperty` sets the validTo property.

Only those classes of information objects, for which class handlers exist, are displayed: By the time of writing only a class handler for local files has been implemented. In addition to TagFS we have implemented a filesystem crawler which tags all files from a given path with tags derived from their directory and file names. This reduces the coldstart problem of not having any tagged resources and makes tagging of new files very easy. Files created later within this directory tree are automatically tagged in the same way.

4. CONCLUSION

We introduced TagFS, a tagging filesystem capable of tagging arbitrary information objects. Future development will focus on feature enrichment, integration with semantic desktop applications and tag dependency analysis based on occurrences and use patterns.

First we will improve the handling of information objects other than physical files and develop views on the history of taggings. We plan to integrate tagFS with Gnowsis[5], a semantic desktop environment. A major drawback of the flat tag space is its size, which can easily comprise some 100 tags. Hence, an important feature is tag clustering in order to reduce the number of tags displayed in one directory to improve usability. Intelligent clustering algorithms could make use of usage statistics of tags and of the relations between tags e.g. through conceptual clustering.

[2]`http://fuse.sourceforge.net/`
[3]`http://www.select-tech.si/fuse/`
[4]`http://www.openrdf.org/`
[5]`http://www.gnowsis.org/`

ACLs in OWL: practical reasoning about access

[Extended abstract]

Norman Gray
VOTech Project, University of Leicester, UK; University of Glasgow, UK
norman@astro.gla.ac.uk

ABSTRACT

The problems of describing, and once described determining, access to resources is one which maps easily to an ontology reasoning problem. We describe a flexible and dynamic access-control system, which naturally supports access-control federation, plus the prototype implementation of a practical system which helps disparate authorities manage and reason about user access to resources.

Keywords

Authorization, OWL

1. INTRODUCTION

Even after the authentication problem has been solved, and it becomes straightforward to identify individuals reliably, we are still left with the authorisation problem, of reasoning about what a given individual is allowed access to. An individual might be allowed access to a resource in their own right, or because they are a member of a collaboration, because their institution is a member of a consortium, because they are located in a particular country, or for some other more elaborate reason. Management of the logic of access is typically distributed, so that the assertion that a particular group has access might be made by the owner of the resource, distinct from the authority who places a specific individual in that group. Add to this the observation that different categories of user might be given partial or otherwise limited access to a resource, and it is clear that managing access control lists (ACLs) is both logically intricate and of considerable interest for a distributed system such as the Semantic Web.

OWL is a very good match to this problem, more so than a rule-based system, since the question of whether a given user should be allowed access to a resource reduces very naturally to, firstly, a question of class subsumption, and secondly the question of whether the user can be deduced to be a member of the class of individuals allowed access.

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006),* Budva, Montenegro, 11th–14th June, 2006

That class can be defined by the owner of the resource, in terms of a variety of other classes expressing institutional affiliations or membership of collaborations. An individual's membership of one of these latter classes can be asserted by a separate authority, and communicated to the resource owner as OWL assertions. Thus the important and taxing problem of federation [1] is more naturally approached from this direction than with other methodologies.

Setting up an ACL ontology in OWL is not a major challenge, though it would need to be as small as possible, and rigourously modular. The more interesting challenge is the practical question of how this functionality may be made available in such a way that asserting authorities may access the reasoning services and manage the sets of assertions conveniently, without necessarily having experience with, or much interest in, the Semantic Web.

Current approaches to this problem depend on the Shibboleth or PERMIS architectures (see [1] for a useful summary). Though carefully designed and implemented, these are designed with a rather static and hierarchical context in mind, and are therefore ill-suited to the more dynamic and fluid relationships of the Semantic Web. Articulating an access policy using an OWL ontology, on the other hand, has the following advantages:

- It is flexible: a very broad range of access policies may be expressed in logical form, since the expression as an OWL ontology is essentially (mobile) code.
- It is secure: it does not have the disadvantage of fully flexible mobile code, since it is a small restricted language, which may be reasoned about reliably.
- The approach can easily build on existing data sets, since an ACL ontology can add semantics to existing LDAP, SAML or other registration sources, reexpressed in RDF.
- Sets of assertions can be composed in a natural and controlled fashion.

In this poster we describe such a system, which we are currently prototyping as a component of the International Virtual Observatory Alliance's (IVOA [2]) security infrastructure.

2. IMPLEMENTATION

We have developed a prototype system which implements this approach.

A resource owner expresses their access policy by defining a class of individuals who are allowed a given access to the resource, such as reading from or writing to it. The owner then defines membership of that class in terms of concepts in this or other ontologies. For example, a university library might allow access to its electronic serials to staff members in that university, plus individuals who have borrowing rights in a partner university. Or a database might be available to researchers in institutions within EU countries. Crucially, the sets of assertions from the partner library, or the geographical information about institutions, can be made available from existing data sources re-expressed in OWL; they are available in discrete packets, so that the trust issues concerning the assertions' provenance and integrity are orthogonal and modular, and can be managed using existing techniques; and the architecture is flexible, requiring only limited coordination between actors, since the resource owner can decide what concepts in the 'foreign' ontology they wish to use to define their allow/disallow classes.

Central to this architecture is a reasoner (which in X.812 terms is a 'Policy Decision Point'). When an individual requests access to the resource (at an X.812 'Policy Enforcement Point'), the reasoner is consulted to determine whether the individual is provably in the class permitted access. In principle this would be an OWL-DL reasoner, but because the relevant ontology would be relatively stable in practice, it could be transformed off-line into a hierarchy which a simpler (and faster) reasoner could use. Confirming the feasibility of this is one of the remaining problems.

We have implemented an initial version of the required functionality in a REST-ful web service called Quaestor, available through a convenient and completely language-neutral API. This generic service manages multiple knowledgebases, composed of sets of client assertions, with the resulting merged ontology queriable through SPARQL. The service is implemented using the Jena and ARQ frameworks, and runs inside the Tomcat servlet engine.

At present (May 2006), the implementation is at a prototype stage. Possible future developments include:

1. embedding the resulting service in a production system;
2. creating simple client applications which assist authorities' authoring of the relevant assertion sets, without obliging users to learn OWL or learn to use SW tools;
3. further refactoring of the access-control ontology to separate generically useful concepts from ones specific to a particular resource;
4. persisting the uploaded models, perhaps using the Manchester Instance Store [3];
5. signing ontologies, so that only certain authorities may update authentication information.

Experience in the coming months, plus confrontation with the use-cases and security infrastructure of the IVOA, will help us determine whether these are indeed in roughly priority order. We acknowledge that task 2 would potentially be a large and challenging task (though it is a path already trodden by the developers of the Gene Ontology [4]), but we expect that there will be a rather large class of simple cases which will need only basic automation, so that it may turn out reasonable for the more complicated, rarer, logic programming tasks to be engineered by hand; finding out how true this is in practice is one of the important goals of our project. Crucially, such clients are only for convenience, and any authority which can in fact generate RDF can interact with the service naturally and directly.

Task 5, though part of a large and important problem in general [5], will be postponable for us, given our overall system design. We expect in any case that it can be factored out from the reasoning aspects of the design.

By the later part of this year we expect to have demonstrated the integration of a service providing SW-style reasoning to a large non-SW architecture.

3. REFERENCES

[1] J Watt, Richard O Sinnott, and A J Stell. Dynamic privilege management infrastructures utilising secure attribute exchange. In *Proceedings of the UK e-Science All Hands Meeting*, 2005.

[2] International virtual observatory alliance [online]. Web: `http://www.ivoa.net`.

[3] Sean Bechhofer, Ian Horrocks, and Daniele Turi. The OWL instance store: System description. In *Proceedings CADE-20*, Lecture Notes in Computer Science. Springer, 2005.

[4] Gene Ontology Consortium. Creating the gene ontology resource: design and implementation. *Genome research*, 11(8):1425–1433, 2001.

[5] XML-Signature requirements [online]. 1999 [cited February 2006]. Web: `http://www.w3.org/TR/xmldsig-requirements`.

Semantic mapping to synchronize data and knowledge bases at the instance level

Olivier Curé
Université de Marne-la-Vallée, Laboratoire ISIS
5, boulevard Descartes
Marne-la-Vallée 77454 France
ocure@univ-mlv.fr

Raphaël Squelbut
Université de Marne-la-Vallée, Laboratoire ISIS
5, boulevard Descartes
Marne-la-Vallée 77454 France
raphael.squelbut@univ-mlv.fr

ABSTRACT

In recent years there has been a growing interest in making existing database (DB) content available to emerging Semantic Web applications. In this paper, a mapping approach between relational DBs and Description Logics (DL) based ontologies has been adopted. Based on the consideration that the DL Abox is a view of the relational DB, we are able to retrieve data en masse from the DBs for integration in the knowledge base (KB). This approach is motivated by the possibility to integrate multiple DBs which may not all be accessible at application run-time. The main contribution of this paper is to tackle the maintenance at the instance level. Thus data inserted, modified or updated in a DB model will be accordingly integrated in the KB model, without a complete processing of the mapping file. A key aspect of this maintenance is the synchronization's efficiency, meaning that integration in the KB of some DB operations may be deferred due to the respect of integrity constraints. This paper focuses on these synchronization issues and their implementation. All synchronization tasks are completely automatized, i.e. human intervention is not required.

1. INTRODUCTION

Several researches are interested in providing ontological engineering tasks such as creation of ontologies and instanciation of the KBs. Most of these solutions are based on the design of expressive and computationally efficient mapping technologies between structured and semi-structured DBs and ontologies. These research solutions usually involve a reverse engineering processing, a task corresponding to the analysis of a "legacy" system in order to identify the system's components and their inter-relationships.

Our system DBOM (DataBase Ontology Mapping) proposes a mapping-based solution for the creation and population of a KB from multiple DBs. But our main contribution is to exploit the mapping file to maintain as synchronized as possible DBs to KB. This synchronization is only considered at the instance level and not at the schema level, meaning that modifications of the DB and ontology schema can not be synchronized between the two models. The maintenance at the instance level are supported by the mapping file and has two flavors (i) a modification of a set of tuples on the DB's side may be reflected on the KB's side, (ii) a modification of a set of concepts and properties instances on the KB's side may be reflected on the DB's side. This paper focuses on the first aspect of the maintenance.

2. RELATED WORK

The primary goal of this paper is to present the maintenance solution of the DBOM system. To our knowledge, there are no researches investigating such an approach. Among related solutions, we distinguish the following categories : (i) creation of a KB (Tbox and Abox) from an existing DB [6, 2, 3], (ii) creation of a DB schema from an existing KB [7], (iii) creation of a mapping between an existing ontology and DB schemata [1, 5], in order to enable information integration. In this approach, an ontology schema corresponding to the DB schema has been manually designed and a mapping is required to enable interoperability. In a nutshell, DBOM belongs to the semi-automatic, like [5, 6], category with loose coupling (data is retrieved en masse from the DBs), like [3, 2] and the target is formalized in OWL DL, corresponding to $\mathcal{SHOIN}(D)$, like [1]. The semi-automatic characteristic is motivated by the fact that DBOM aims, but is not limited, to develop light ontologies supporting inferences in domain-specific applications. By light ontologies we mean KBs that only contain data involved in reasoning activities. These characteristics make DBOM similar to D2R MAP but with the ability to integrate multiple data sources [4]. However another important difference between these two solutions is in the terminological axiomatization possibilities of DBOM which enable the creation of ontologies as expressive as OWL DL. Finally, DBOM proposes additional services one of which, maintenance solutions, is emphasized in the rest of this paper.

3. DBOM FRAMEWORK

3.1 Overview

DBOM is based on the use of a declarative mapping, serialized in XML, which is a set of explicit correspondences between components of the DB and KB models. The processing of the mapping file enables to create a TBox and instantiate the ABox, considering it as a view of the relational DB. Our contribution to this issue lies in the possibility to richly axiomatize the terminology; thus permitting the creation of expressive ontologies. But the most interesting contribution is the solution proposed to maintain the synchronization between the DB tuples and the Abox. This synchronization is based on automatically created, at mapping processing time, SQL triggers which are fired whenever a "write" query (meaning insert, delete or update SQL queries) is processed on a DB relation used in the mapping file. These triggers are calling Java methods developped within our framework which are responsible for the update of the Abox.

We have developed a Protégé plug-in version of DBOM. This plug-in aims to simplify the creation of mapping files using a graphical user interface and all the features provided by Protégé, principally from the OWL plug-in. We now refer to "members" of the mapping as the set of concepts and object properties. We make the distinction between concrete and abstract members. The comprehension of concrete and abstract members is relatively straightforward as it is equivalent to the assumption made in Object-Oriented Programming. Thus instances (individuals) can be created for a concrete concept and a concrete object property can relate two existing individuals. Abstract members can not be instantiated and they aim to design a hierarchy of members where final (leafs in a tree representation) members should be concrete. The DBOM Protégé plug-in is efficiently integrated in the Protégé framework to enable the design of KBs. In the nutshell, the DBOM plug-in aims to compose concrete members and their SQL queries, via interactions with the mapped DBs presented as a tree. All over ontological tasks can be performed via the OWL tabs, i.e. axiomatizations.

In the following example, we highlight a possible mapping of a relational schema to a TBox.
Relational schema :
person (<u>idPerson</u>, name, idGender)
gender (<u>idGender</u>, name)
The mapping file defines the following TBox (*Person* is an abstract concept).
$Man \sqsubseteq Person$
$Woman \sqsubseteq Person \sqcap \neg\, Man$
The queries of the concrete concepts (*Man* and *Woman*) are presented in the form of conjunctive queries.
$Man \equiv \{ (X,Y) \mid person(X,Y,Z) \wedge gender(Z,U) \wedge U=\text{'male'}\}$
$Woman \equiv \{ (X,Y) \mid person(X,Y,Z) \wedge gender(Z,U) \wedge U=\text{'female'}\}$

This plug-in solution enables to load an existing OWL KB and add new members via the integration with DBs.

3.2 Synchronization issues

A central aspect of the instantiation is the "membership determination solution" which aims to find the appropriate concrete member to create, modify or delete given the firing of a DB trigger. This algorithm enables to detect that either one of the instances of the *Man* or *Woman* concrete concepts of example 2 can be updated from the firing of a trigger on the person relation.
The main issue of the synchronization lies in the respect of integrity constraints (ICs) defined in DB sources. In this paper, we are concerned with the most relevant ICs encountered in relational DBs : key, foreign key and functional dependencies. These ICs force the system to postpone some of the instantiations in the ABox. Thus, a complete synchronization is ensured from the processing of required SQL queries in the DBs. The management of delays has to be taken care of by DBOM's synchronization policy. Four different stages can be encountered : (i) no action, meaning that the "write" queries has no effect on KB instances, (ii) simple action, meaning that a unique object is treated in the KB, (iii) multiple action, meaning that several objects can be treated due to the feedback effect of synchronisation (one action can cause many postponed actions), (iv) postponed action, meaning that no action can be processed but the system is left in a state where future triggers may fire multiple actions. DBOM also supports SQL referential actions for update and delete rules, i.e. cascade, set null, set default and no action. We now consider referential actions as specialized triggers for automatically maintaining referential integrity in DBs.

4. CONCLUSION AND FUTURE WORKS

The DBOM system is implemented using the Java language and Hewlett-Packard's Semantic Web Jena framework and Protégé. Tests have been conducted with PostgreSQL and our DBOM plug-in. The addition of terminological axioms in the DL Tbox enables to highlight inconsistencies on the KB that could not be detected on the DB instance. We are currently working on the detection of the inconsistencies as well as their explanations to end-users. This issue also broadens our approach to DB repair and the view-update problem.

5. REFERENCES

[1] An Y., Borgida, A., Mylopoulos, J. : *Inferring Complex Semantic Mappings Between Relational Tables and Ontologies from Simple Correspondences.* Proceedings of OTM Conferences (2005) 1152-1169.

[2] Bizer, C. : *D2R MAP - A database to RDF Mapping Language.* The Twelfth International World Wide Web Conference (2003).

[3] Borgida, A. : Loading Data into Description Reasoners. Proceedings of ACM SIGMOD International Conference on Management of Data (1993) 315–333

[4] Curé, O., Squelbut, R. : *Data integration targeting a drug related knowledge base* To appear in the proceedings of EDBT 2006 Information Integration and Health Applications Workshop.

[5] Handschuh, S., Staab, S., Volz, R. : On Deep Annotation. 12th International World Wide Web Conference (2003). 431–438.

[6] Stojanovic, L., Stojanovic, N., Volz, R. : *Migrating data-intensive web sites into the semantic web.* Proceedings of the ACM Symposium on Applied Computing SAC (2002) 1100–1107

[7] Moreno Vergara, N., Navas Delgado, I., Francisco Aldana Montes, J. : *Putting the Semantic Web to Work with Database Technology.* IEEE Data Engineering Bulletin Volume 26, issue 4 (2003) 49–54

Community-based Annotation of Multimedia Documents

Ajay Chakravarthy
University of Sheffield
Regent Court, 211 Portobello Street
Sheffield, S1 4P, United Kingdom
+44-114-2221945
a.chakravarthy@dcs.shef.ac.uk

Vitaveska Lanfranchi
University of Sheffield
Regent Court, 211 Portobello Street
Sheffield, S1 4P, United Kingdom
+44-114-2221945
v.lanfranchi@dcs.shef.ac.uk

Fabio Ciravegna
University of Sheffield
Regent Court, 211 Portobello Street
Sheffield, S1 4P, United Kingdom
+44-114-2221945
f.ciravegna@dcs.shef.ac.uk

ABSTRACT
In this paper, we analyse the process of annotating multimedia documents inside a community as a way to enable knowledge sharing and reuse.

Keywords
Semantic annotation interfaces, multimedia enrichment, cooperative annotations, knowledge sharing.

1. INTRODUCTION
Nowadays a large and growing amount of information is stored in various multimedia formats, such as images, video, audio and it is therefore important to discover a way to manage this information. Previous research in personal image management [7, 8] and text annotation [3, 5, 6, 9] demonstrated how annotating images or documents could be a way to organise information and transform it into knowledge that can be retrieved easily later. Literature also details how manual annotation can be a labour intensive and tedious task [3]. Several tools already use metadata, free-text annotations and ontology-based annotations to enrich images and documents with added knowledge [3, 5, 6, 7, 8, 9], but a step further can be reached by providing methods to reuse the produced knowledge.
By sharing annotations inside a community of users can make the annotation task easier and facilitate the reuse of annotated knowledge, suggestions and recommendations could be provided on the basis of other users annotations. Previous research has investigated collaborative annotation systems for mobile phones [2] and for personal images collections [4]. The aim of our project is to offer users an interface for performing multimedia document enrichment in a shared environment, while using intelligent techniques to provide suggestions.

1.1 How is the annotation performed?
As previous research demonstrated there are various ways of annotating documents, especially images, when doing different tasks: (1) There are numerous business users for which precise image annotation is very important, for example patent agents, journalists, professional image researchers or simply users inside an organisation (E.G. for Knowledge Management). Typically users require not only an accurate description of the image characteristics (i.e. shutter speed, date and so on) but also domain specific knowledge (patent number, location, product ID, etc.). In this scenario, the annotations should be performed in a standardised and constrained manner, typically using a strict ontology to ensure that the information is consistent and can be shared among the user's community. (2) In other cases, the strict standardisation and constraints are less relevant, for example users that simply want to annotate their own pictures for sharing with friends: in this case the social dimension is more important; there is no need of formal classification but more of a way to attach emotions and memories, to be able to retrieve them in the future. In these cases free-text annotation prove to be more interesting for the users [7].

Demos and Posters of the 3rd European Semantic Web Conference(ESWC 2006), Budva, Montenegro, 11th - 14th June, 2006

Another important consideration is to identify and make explicit relations between annotated instances both within a single document and external resources. Relations can be cross-media, so they may interrelate, for example, text with part of an image: "engine1000" mentioned in the text could have a relation, for example, "has_part" with an instance "turbine" identified in a picture.

In our project we aim to conciliate these approaches providing an interface for adding ontology-based, free-text and relational annotations within multimedia documents. This conciliation of approaches allows a user to perform any desired annotation for any required task.

1.2 Sharing annotations in a community
The idea of sharing annotations inside a community is particularly powerful if thought in terms on Knowledge Management (KM). In a KM perspective, the knowledge must be shared to maintain the organisational memory through time [1].

We propose to use semantic annotation inside KM applications to ease the explicitation of the implicit knowledge. The process of annotating a document should start from a shared ontology that is agreed between the members of the organisation and will be used as a common vocabulary to annotate the documents in a consistent way. Moreover free-text annotations can be used to add personal comments or opinions on the document in general or on the annotations and relations can be identified between the found instances and commented. If the annotated documents are then shared inside this community also the attached annotations could be. This sharing facilitates the annotation task and makes other users aware of comments and opinion that may be particularly relevant and otherwise would get lost

2. Community Annotation in AKTiveMedia

AKTiveMedia is a user centric system for multimedia document annotation, developed at the University of Sheffield. AKTiveMedia allow users to annotate textual, image or multimedia documents in a collaborative way, sharing their experience with other members of the community. Language technologies and a web service architecture are adopted to provide a context specific suggestion mechanism: for example when the user is annotating a region of an image as a "part" of an engine, the system suggests all the possible parts present in the ontology or in other user annotations for that engine and the user can select the right one. The same happens for relations, again inferred from the ontology and the knowledge base and suggested to the user on the base of the concept selected: for example, when the part has been chosen, the user can select a "has_fault" relation and drag and drop the text in the document that describes the fault; when they are inserting a free-text for describing a fault, the system offers suggestions based on what other users previously input. Sharing in AKTiveMedia is possible due to the use of a two steps persistence model to save the annotations: 1) when the user annotates a document, the annotations are fist saved in a local repository, 2) then they are imported by an automated web service into a central repository. This operation is repeated at regular intervals (see Figure 1).

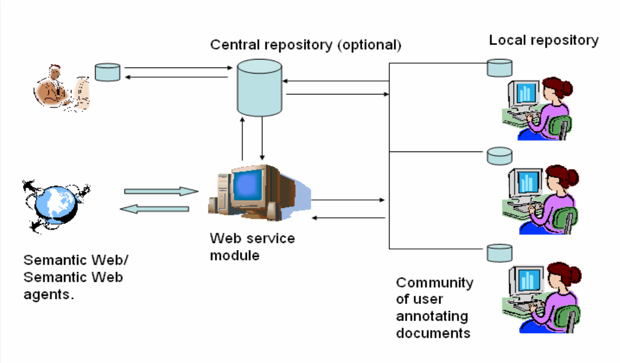


Figure 1 - AKTiveMedia two step persistent architecture

While the user is annotating a document, the system automatically performs an intelligent query on the central repository that contains the annotations knowledge base and extracts possible annotations and comments previously inserted by other members of the community and suggest them to the user. This mechanism can be used in two different ways in the annotation process: (1) to suggest a new annotation inside an image or a document; (2) to auto-complete the fields when the user is typing an annotation. Offering an auto-completion service enables the user to choose from alternatives already inserted by other users, thus preserving consistency while annotating. Suggesting new annotations and related documents is instead a way to make explicit connections and ideas that were not known to a user, taking advantage from the experience of the community. The produced knowledge is also used as a way to establish connections with and to navigate the information space: when the user annotates a part of an image as "sand-damage" upon a "turbine" the system uses those annotations to retrieve other related images and document. We believe this process could facilitate knowledge sharing inside a community but could also create privacy problems about the visibility of annotations and comments. These privacy issues are addressed by marking the annotations as public, private or restricted, thus presenting them only to users with the right to access them.

3. ACKNOWLEDGMENTS

This work was carried out within the AKT project (http://www.aktors.org), sponsored by the UK Engineering and Physical Sciences Research Council (grant GR/N15764/01).

4. REFERENCES

[1] Ackerman M., McDonald D. Answer Garden 2: Merging Organizational Memory with Collective Help. In Proceedings of the Computer Supporting Cooperative Work Conference, CSCW 1996, Cambridge MA USA, 97-105

[2] Appan, P., Shevade, B., Sundaram, H., and Birchfield, D. 2005. Interfaces for networked media exploration and collaborative annotation. In Proceedings of the 10th international Conference on intelligent User interfaces (San Diego, California, USA, January 10 - 13, 2005). IUI '05. ACM Press, New York, NY, 106-113.

[3] Ciravegna F., Dingli A., Petrelli D. and Wilks Y. User-System Cooperation in Document Annotation based on Information Extraction in Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02), 1-4 October 2002 - Siguenza (Spain), Lecture Notes in Artificial Intelligence 2473, Springer Verlag

[4] Davis M., Good N. and Sarvas R. From Context to Content: Leveraging Context for Mobile Media Metadata, MobiSys 2004 Workshop on Context Awareness, Boston, MA, 2004.

[5] Handschuh S., Staab S., and Ciravegna F.. S-CREAM- Semi-automatic CREAtion of Metadata. In Proceedings of the 13th International Conference on Knowledge Engineering and

[6] Kahan J., Koivunen M., Prud'Hommeaux E., and Swick R.. Annotea: An Open RDF Infrastructure for Shared Web Annotations. In Proc. of the WWW10 International Conference, Hong Kong, 2001.

[7] Kang, H. and Shneiderman, B. Visualization Methods for Personal Photo Collections: Browsing and Searching in the PhotoFinder In Proc. IEEE International Conference on Multimedia and Expo (ICME2000), New York City, New York.

[8] Kuchinsky A., Pering C., Creech M.L., Freeze D., Serra B., Gwizdka J., FotoFile: A Consumer Multimedia Organization and Retrieval System, Proceedings of ACM CHI99 Conference on Human Factors in Computing Systems, 496-503, 1999.

[9] Vargas-Vera M., Motta E., Domingue J., Lanzoni M., Stutt A., Ciravegna F.: MnM: Ontology driven semi-automatic or automatic support for semantic markup. In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02. Springer Verlag, 2002

Web Service Discovery – A Reality Check

Daniel Bachlechner
Digital Enterprise Research Institute
Innsbruck, Austria
daniel.bachlechner@deri.org

Katharina Siorpaes
Digital Enterprise Research Institute
Innsbruck, Austria
katharina.siorpaes@deri.org

Holger Lausen
Digital Enterprise Research Institute
Innsbruck, Austria
holger.lausen@deri.org

Dieter Fensel
Digital Enterprise Research Institute
Innsbruck, Austria and Galway, Ireland
dieter.fensel@deri.org

ABSTRACT
Web services are about the integration of applications via the Web. Hereby, the programming effort should be minimized through the reuse of standardized components and interfaces. Semantic Web services try to provide the next step through mechanizing important sub tasks within a service-oriented architecture. Otherwise, significant manual programming effort would remain as a bottleneck for this approach. One of the sub tasks in a service-oriented architecture is service discovery. While a significant number of papers have already been published in this area, most of them are more concerned in providing yet another illustration for an arbitrary logical framework rather than providing a contribution that meets current constraints in given practical settings. On the poster, we provide a comparison of existing approaches towards Web service discovery based on empirical findings. This sets the basis for analyzing the strengths and weaknesses of the existing approaches as well as the prediction of future potential improvements in this area. We also identify a useful role for semantic techniques as long as it is in a proper setting.

Categories and Subject Descriptors
D.2.11 [**Software Engineering**]: Software Architecture – *data abstraction, domain-specific architectures, information hiding, languages, patterns.*

General Term
Measurement, Experimentation

Keywords
Semantic Web Services, Discovery, Service-oriented Architecture

1. INTRODUCTION
Service-oriented architectures (SOA) emphasize that it is the service that counts for the customer, not the specific software or hardware component that is used to implement it. SOAs will likely become a leading software paradigm quickly. However, they will not scale without significant mechanization of service discovery, service adaptation, negotiation, service composition, service invocation, and service monitoring; as well as data, protocol, and process mediation [5]. Web services are service endpoints in such architecture. If the SOA paradigm succeeds there will be soon several thousand services, which can be used for composing required applications. However, for this, these services must first be discovered. Within the semantic Web community, many of the publications on service discovery tend to place more emphasis on certain aspects of reasoning rather than on focusing on current constraints and foreseeable evolvement of service discovery (cf. [1], [2]). The survey summarized on the poster takes the opposite approach. We enumerate existing approaches for public Web service discovery, compare them with respect to specific criteria and identify their strength and weaknesses. Based on the evaluation's results we conclude potential paths for semantics in Web service discovery as an extension of current approaches.

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 11th - 14th June, 2006.

2. SURVEY
Based on previous work [4] we have identified several approaches for Web service discovery that are actually deployed and exceed the scope of a dozen test services. We have examined the standard UDDI registry approach, search via specialized portal sites and customized searches using standard Internet search engines.

2.1 Current Approaches
The first of the three described approaches for current Web service discovery is based on UDDI. UDDI (Universal Description, Discovery, and Integration) is a standard for centralized repositories. The first UDDI Business Registry (UBR) nodes were run by IBM, Microsoft, SAP and NTT Com.

Examples of the second approach are specialized portals which gather services using focused crawlers as well as manual registration. The list of Web service engines investigated within the scope of the study includes XMethods, BindingPoint, WebServiceX.NET, Web Service List, StrikeIron, Woogle, RemoteMethods, and eSynaps. This list of engines includes to our knowledge all relevant portals of the time of writing. Some repositories known from previous work like SalCentral and Grand Central could not be accessed during the time of the survey and hence have not been evaluated.

The third approach uses standard Web search engines which are able to restrict the search to WSDL files. We analyzed the search engines Google and Baidu with respect to their ability to facilitate and enable Web service discovery. Google and Baidu have different means to restrict search queries to specific types of documents, and given the huge size of the underlying document index, both likely to be big players in the long-run.

2.2 Criteria

The criteria used for the evaluation can be classified into two groups, the first of which represents basics for Web service discovery and deals with core criteria like the ways of how a search can be conducted, number of available services, status information, and supported interfaces. The second criteria group consists of service rating, test and demo features (like WSDL analyzer), and service costs which allows service discovery to be more precise and less time-consuming.

3. SURVEY RESULTS

The Web service resources presented on the poster follow many different approaches of service discovery with varying success. The findings of the evaluation are represented in tabular form. The table describes the current approaches in Web service discovery in terms of the introduced evaluation criteria. The three groups of resources as well as the two groups of criteria are clearly separated. We used the developed evaluation scheme to conduct an in-depth comparison of the discovery approaches.

The majority of the approaches relies on keyword search as well as category browsing whereas XMethods only shows services in a simple list format. The UBR also allow searching for service providers and tModels. The Web service search engine Woogle additionally offers template search on operations. Obviously the state of the art of search functionality is rather limited and hampers usability. Semantic Web services could enable a more comprehensive search as well as automation of tasks. Especially in the UBR, the location of Web services is difficult as it does not provide an efficient interface for querying services. The name of a Web service, a Web service provider or a tModel must be known to get further details. The UBR keyword search only takes names into account and ignores service descriptions. Considering service descriptions could be promising in theory, but unfortunately most of the descriptions available are deficient and of low quality. Due to the limited extent of human readable descriptions in the UBR, discovery is a cumbersome and time-consuming effort. Most numbers provided by Web service search engines, concerning the number of registered Web services, are vague and imprecise. However, it is obvious that Google provides a significantly higher number of WSDL files.

The number of available services in terms of a specific discovery provider is an important indicator for the comprehensiveness of a Web service discovery engine. However, at large, service functionality and quality are of course far more important than quantity. Some Web service resources provide functionality to determine whether a service is active or not. The UBR does not provide any status information at all while StrikeIron and Woogle display the status of a listed Web service (active or inactive). BindingPoint allows for excluding inactive Web services from its listings. Another helpful piece of information provided by BindingPoint in this context is the average response times of specific services. All evaluated resources for locating Web services have Web interfaces. Selected ones also provide SOAP and UDDI Private Registry interfaces as well as RSS feeds, WS-Inspection.

4. CONCLUSIONS

Based on our findings, searching with Google has the best coverage, although the precision is limited since there is no single way to restrict a search to only retrieve active and working services supposed to test examples. Most of the public UDDI registries have been discontinued in early 2006, however, due to the limited quality of the contained data, for searching public services they have never been a good source. All existing specialized Web search engines provide less coverage than Google. However, the standard model of Google is not well suited for Web service discovery. Neither the identification of potential services through pure key word extraction nor the relevance ranking based on HTML characteristics such as hyperlinks and title tags provides much of a use in a Web service scenario. The usage of standardized vocabulary such as UNIFACT or eClass to classify Web services could significantly improve the correctness and completeness and do not provide much of a burden to Web service providers. If needed, this task can be mostly automated by approaches such as GoldenBullet [3]. Furthermore the page ranking mechanism of Google that uses the link structure and special properties of HTML documents are not applicable to WSDL files. Therefore, different post processing and filtering mechanisms of the output of Google are needed. This is a task where richer semantic annotations can play a role.

Simple application of IR technologies and later the use of ontologies to describe standard vocabulary are the most promising approaches for the near future. Rich formal frameworks are required as well, but should be considered more in the scope of semi closed environments (e.g. extra nets), where full automation is possible. For the near future the role of central portal providers will most likely become more important in the domain of public services: Take Amazon as an example. The effort in maintaining and developing this central repository is high but it is profitable too. StrikeIron for instance follows a similar business model. However in the long run the business model might be invalidated by the advancement of technology: If semantic Web service technologies advance it is likely that such intermediates (between service provider and consumer) will loose its current importance.

5. REFERENCES

[1] Akkiraju, R., Goodwin, R., Doshi, P. and Roeder, S.: A method for semantically enhancing the service discovery capabilities of UDDI. In S. Kambhampati and C.A. Knoblock, eds., *Proceedings of the IJCAI-03 Workshop on Information Integration on the Web*, pages 87-92, 2003.

[2] Benatallah, B., Hacid, M-S., Rey, C. and Toumani, F.: Request rewriting-based Web service discovery. In *The Semantic Web - ISWC 2003*, pages 242-257, October 2003.

[3] Ding, Y., Korotkiy, M., Omelayenko, B., Kartseva, V., Zykov, V., Klein, M., Schulten, E. and Fensel, D.: GoldenBullet in a Nutshell. In *Proceedings of the 15th International FLAIRS Conferenc*e, Pensacola, USA, 2002.

[4] Fan, J. and Kambhampati S. A Snapshot of Public Web Services, *SIGMOD Record*, 34(1), pages 24-32, March 2005.

[5] Fensel, D. and Bussler, C.: The Web Service Modeling Framework WSMF. In *Electronic Commerce Research and Applications*, 1(2), pages 113–137, 2002.

Ontology Access in Grids with WS-DAIOnt and the RDF(S) Realization

Miguel Esteban Gutiérrez
Ontology Engineering Group
Campus de Montegancedo s/n
28660, Boadilla del Monte, Madrid Spain
+34913363670
mesteban@fi.upm.es

Asun Gómez-Pérez
Ontology Engineering Group
Campus de Montegancedo s/n
28660, Boadilla del Monte, Madrid Spain
+34913367439
asun@fi.upm.es

Óscar Muñoz García
Ontology Engineering Group
Campus de Montegancedo s/n
28660, Boadilla del Monte, Madrid Spain
+34913363670
omunoz@delicias.fi.upm.es

ABSTRACT

This paper presents WS-DAIOnt, a framework for defining ontology access service interfaces in terms of the WS-DAI specification, extending it with the patterns, properties and behaviours needed for providing ontology access in a Grid environment. We also present WS-DAIOnt-RDF(S): a realization of WS-DAIOnt for accessing RDF(S) ontologies.

Categories and Subject Descriptors

D.2.0 [**General**]: Standards, H.4.m [**Information Systems**]: Miscellaneous

General Terms

Design, Standardization.

Keywords

Grid computing, ontologies, RDF(S).

1. INTRODUCTION

The increasing use of semantic technologies has reached almost all computer science related fields, including the Grid computing. The next generation Grid should make virtual the notion of distribution in computation, storage, and communication over unlimited resources with well defined computational semantics. A Grid node may provide new services, functions, and even new concepts that are now unknown to clients. The semantics of such services are defined by means of Ontologies, which are formal and explicit specifications of a shared conceptualization.

Ontologies can be used in the Grid for several purposes: for describing policies and sharing information, services and computing resources in Virtual Organizations; for describing formal and non formal properties of Grid resources and services; for accessing data catalogs in a conceptual and multidimensional way, etc. Right now, few Grid applications use ontologies but the access to these ontologies is not integrated in the Grid infrastructure, and the current OGSA architecture [1] does not consider ontology usage, does not define protocols and does not tackle this issue. Therefore, the provision of proven value mechanisms for accessing and managing ontologies in Grid environments is the main priority if the Grid wants to make profit of the semantic technologies already available in other areas such as the Semantic Web.

The Semantic Grid community [2] does not start from scratch. The Semantic Web community has already developed languages and tools for building and using ontologies. The W3C has recommended three languages to be used for implementing ontologies in the Semantic Web (RDF(S) and OWL) and several ontology development tools (i.e., Protégé, WebODE, KAON) for supporting the creation of ontologies in such languages. The languages recommended differ in their expressiveness (the kind of knowledge that can be represented) and in their inference mechanisms (the kind of reasoning they carry out). However, the diversity of ontology languages and tools causes translation problems, which appear when an ontology developer decides to reuse anontology with a tool/language different than the one used in its development. At the same time, several APIs can access ontologies implemented in a given language and the ontology user should know how to retrieve the ontology content, i.e. the Sesame, Jena or 3Store APIs in the case of RDF(S). At present, the Semantic Web community does not have a standard mechanism or protocol for accessing ontologies implemented in a given ontology language in a storage-and-retrieval system independent fashion, and this leads to severe interoperability problems.

In order to be able to apply semantic technologies in the Grid, we must first face and solve these interoperability issues. Thus, to provide the appropriate means for accessing and using ontologies in the Grid is a crucial issue if semantic technologies are to be used, as it is crucial the transition from monolithic, centralized ontology services to a virtual organization of Grid compliant and Grid aware ontology services that can coordinate and cooperate with each other to progress towards the Semantic Grid

2. WS-DAIOnt: a framework for specifying ontology access services

One of the main goals of the OntoGrid project is to explicitly share and deploy knowledge to be used for the development of innovative Grid infrastructure and for Grid applications. To address this challenge, the OntoGrid project is developing a Semantic Grid reference architecture (named Semantic OGSA, a.k.a. S-OGSA) and the technological infrastructure for the rapid prototyping and development of knowledge-intensive distributed open services for the Semantic Grid. A key module of this Semantic OGSA reference architecture is the component that provides access to Ontologies, being our main goal in OntoGrid to develop, build on, adapt and extend existing ontology services to be Grid compliant. This high level, very general objective can be refined as follows: to provide seamless access to heterogeneous and distributed ontology sources created according to different knowledge representation formalisms by means a uniform ontology access mechanism.

With the goal of avoiding the proliferation of different access mechanisms for ontologies implemented in languages of the Semantic Web, the OntoGrid project is specifying and designing an ontology access mechanism for the Grid, whose formal name is WS-DAIOnt. It provides a WS-DAI [3] based framework for defining ontology access service interfaces; it uses the standard grid data access vocabulary, and extends the data access mechanisms with the patterns, properties and behaviours needed for providing ontology access. The WS-DAIOnt specification and the accompanying realizations (WS-DAIOnt-RDF(S), …) define the data access services that are needed for dealing with ontologies in Grid environments. The specification is fully compliant with S-OGSA (so with OGSA) and WS-DAI and is based on up-to-date Web Services standards such as WS-RF and WS-Addressing.

WS-DAIOnt is built over four pillars:

- *Unified basic terminology*. WS-DAIOnt defines a neutral vocabulary for naming the ontology elements to be used when dealing with ontologies in Grid environments.
- *Ontology elements usage patterns*. WS-DAIOnt defines how the messages, methods, interfaces, and services must be specified in order to provide functionalities in a standard way.
- *Ontology elements possible relationships*. WS-DAIOnt defines how to specify how each ontology element is related to each other.
- *Ontology access services behaviors*. WS-DAIOnt defines the expected behaviour of the predefined common components and functionalities.

By extending WS-DAI and, therefore, the OGSA architecture, with WS-DAIOnt and the accompanying realizations, we provide the current Grid architecture with a standard way for supplying ontology access and management capabilities, thus enabling the future integration of semantic technologies in the Grid architecture. A preliminary detailed version of the WS-DAIOnt specification is available in the deliverable D3.1 of the OntoGrid project [4], available at the project's site: http://www.ontogrid.net.

3. WS-DAIOnt-RDF(S): accessing RDF(S) in the Grid

To test the correctness of our approach, we decided to start the implementation of WS-DAIOnt with ontologies implemented in RDF(S). The WS-DAIOnt-RDF(S) realization offers a framework for defining ontology access service interfaces using the WS-DAI vocabulary and for defining the set of messages, properties and behaviors needed to provide ontology access to ontologies implemented in RDF(S) (see figure below)

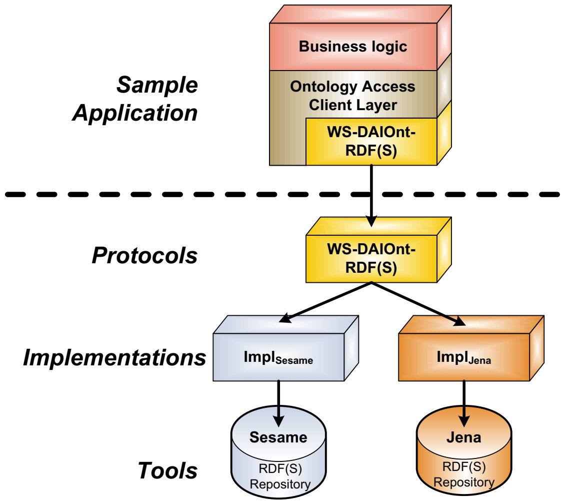
The infrastructure chosen for developing WS-DAIOnt-RDF(S) is the Globus Toolkit 4, which provides the Grid infrastructure, specifically the Java WS-Core. Regarding the ontology repository, we selected Sesame, using SeRQL for accessing the ontologies stored in it.

In the WS-DAIOnt-RDF(S) framework architecture we can distinguish three service layers: the upper layer allows selecting the repository to be used; the intermediate layer allows interacting with the selected repository directly; finally, the lower layer includes the services that permit interacting directly with the RDF(S) knowledge components. Up-to-date the following services have been fully developed:

`RDFSRepositorySelectorService`. This service allows selecting the specific repository to be used in case that multiple repositories are available. Each repository is identified by a unique identificator.

`RDFSRepositoryService`. This service provides access to all the knowledge components (properties, statements, classes, etc.) in the repository. Some of the operations provided are, for instance, to get all the classes of the repository (`getAllClasses`) or to get a class by its URI (`getClass`).

`RDFSClassService`. This service provides access to a given RDFS class, i.e., get the sibling classes of the class (`getSiblings`), get the related subclasses (`getSubClasses`) or superclasses (`getSuperClasses`).



T

The RDFSClassService needs to interact directly with the Sesame repository to execute its methods. As we are trying to provide storage-independent access to RDF(S), we have to decouple the repository from the service implementation.

Generally speaking, those services which have to interact with the repository must do it in a loosely coupled way. This is achieved by an extra data access layer, namely RDFSConnector, which adapts the access to the repository to the services requirements.

4. ACKNOWLEDGMENTS

This work is supported by the OntoGrid project (FP6-511513) and by a U.P.M. PhD student grant.

5. REFERENCES

[1] I. Foster (Editor), D. Berry, A. Djaoui, A. Grimshaw, B. Horn, H. Kishimoto (Editor), F. Maciel, A. Savva (Editor), F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich; The Open Grid Services

[2] D. De Roure, N.R. Jennings, N.R. Shadbolt; The Semantic Grid:Past, Present and Future. In: Proceedings of the IEEE Volume 93, Issue 3, March 2005, pp 669-681.

[3] M. Antonioletti, M. Atkinson, S. Malaika, S. Laws, N. W. Paton D. Pearson and G. Riccardi; Web Services Data Access and Integration (WS-DAI). DAIS-WG Informational Draft, 13th Global Grid.

[4] M. Esteban Gutiérrez (ed), S. Bechhofer, O. Corcho, M. Fernández-López, A. Gómez-Pérez, and more; Specification and Design of Ontology Grid Compliant and Grid Aware Services. D3.1 OntoGrid Project, 27th April 2005.

An ontology-based system for interactive exploration of information sources

Jeroen Wester
jeroen@aduna-software.com

Christiaan Fluit
chris@aduna-software.com

Jeen Broekstra
jeen@aduna-software.com

Herko ter Horst
herko@aduna-software.com

Aduna
Prinses Julianaplein 14-B
3817 CS Amersfoort
The Netherlands
+31 (0)33 465 99 87
http://www.aduna-software.com/

Arjohn Kampman
arjohn@aduna-software.com

ABSTRACT

We outline an ontology-based architecture on three levels that enables users to explore information sources. On the data level Aperture is a Java framework for extraction of data and metadata. On the model level extracted metadata is stored in an RDF framework called Sesame. On the presentation level a faceted navigation engine called Spectacle and Cluster Map visualization enable ontology-based interaction with the data.

Keywords

Sesame, Aperture, Spectacle, Cluster Map, facet navigation, information visualization, browsing.

1.INTRODUCTION

This poster presents an ontology-based architecture that enables users to explore information sources (Figure 1). The architecture has three levels: data, model and presentation. The core of the system is a RDF database for storage and querying of information sources. On the data level, data and metadata are extracted from information sources. Information visualization and faceted navigation components, on the presentation level, enable the user to interact with the data and metadata. In the following sections, we briefly introduce each level and the software components that enable it in our proposed architecture.

2.DATA LEVEL

Aperture 6 is a Java framework for extracting and querying full-text content and metadata from various information systems (e.g. file systems, web sites, mail boxes) and the file formats (e.g. documents, images) occurring in these systems. The framework enables crawling heterogeneous information sources and extracting metadata from these information sources. Aperture makes use of RDF graphs to communicate information between components. Aperture uses a dedicated RDF vocabulary for describing properties of documents, files, and e-mails that is a specialization/extension of the Dublin Core metadata vocabulary.

3.MODEL LEVEL

At the model level, information retrieved by Aperture's metadata extractors is stored and made available to higher-level, presentation components, using a Sesame RDF repository 6. The Sesame framework provides the system with a flexible, scalable way of storing and manipulating large RDF models, and can provide reasoning support for RDF Schema- or OWL-based ontologies.

Using Sesame's support for declarative querying in SeRQL 6 or SPARQL 6, information expressed in Aperture's own RDF vocabulary can be mapped or transformed to other vocabularies.

In the case of the current architecture, SeRQL querying is used to map the information to a vocabulary that is especially suited for use by the Spectacle faceted navigation engine.

4.PRESENTATION LEVEL

At the presentation level two components, a faceted navigation engine called Spectacle 6 and an information visualization component called the Cluster Map library 6, enable the user to interact with the retrieved data and metadata by querying the RDF database.

The Spectacle engine allows users to navigate an information space by progressively selecting desired facet values of information objects.

The Cluster Map is an information visualization technique for sets of classified objects. Its main purpose is to show if and how these sets overlap, very similar in nature to Venn diagrams and Euler diagrams.

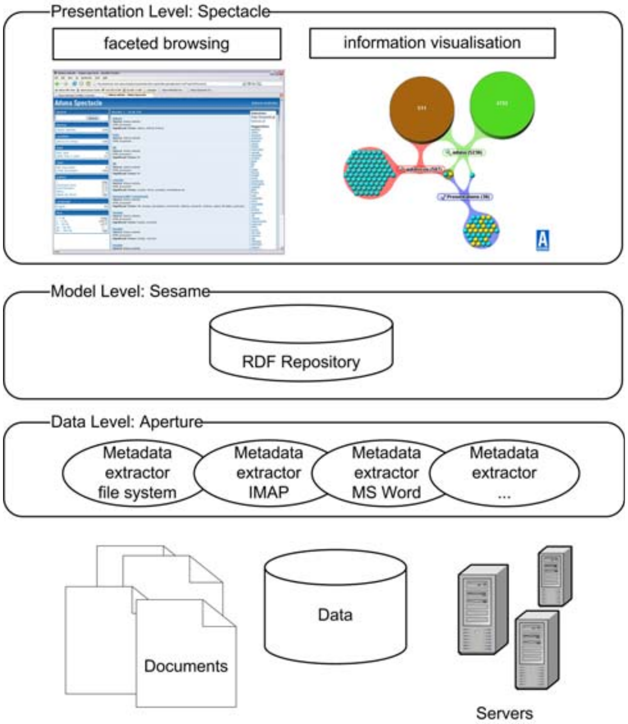


Figure 1: overview of the system architecture

5.THE ROLE OF ONTOLOGY

The ontology-based nature of this architecture is based on the following choices:

a. The crawling of information sources on the data level is ontology-driven. This means that Aperture will crawl for metadata that is part of a defined vocabulary. For example, it will crawl Dublin Core-like properties such as title and author and expose that information as an RDF graph.

b. The interaction on the presentation level is ontology-driven. This means that users interact with the model and the data while searching, browsing or exploring. For example, the Spectacle interface can present the user a 'Document type' facet that corresponds to the class 'Document Type' in the underlying ontology, listing known instantiations of that class as navigation steps.

6.REFERENCES

[1] Seaborne, S. and Prud'hommeaux, E. *SPARQL Query Language for RDF.* W3C Candidate Recommendation, April 6 2006. See http://www.w3.org/TR/rdf-sparql-query/.

[2] Becket, D. and Broekstra, J. *SPARQL Query Results XML Format*. W3C Candidate Recommendation, April 6 2006. See http://www.w3.org/2001/sw/DataAccess/rf1/.

[3] Broekstra, J., Kampman, A. and van Harmelen, F. Sesame: An Architecture for Storing and Querying RDF and RDF Schema. In *Proceedings of the First International Semantic Web Conference (ISWC 2002)*, Sardinia, Italy, June 9-12 2002, p. 54-68. Springer-Verlag Lecture Notes in Computer Science (LNCS) no. 2342. See also http://www.openrdf.org/.

[4] Broekstra, J.: SeRQL: A Second-Generation RDF Query Language. Chapter 4 in *Storage, Querying and Inferencing for Semantic Web Languages*. PhD Thesis, Vrije Universiteit Amsterdam (July 2005). ISBN 90-9019-236-0. See also http://www.openrdf.org/doc/SeRQLmanual.html.

[5] The Aperture Framework. See http://aperture.sourceforge.net/

[6] The Aduna Cluster Map Library. See http://aduna.biz/products/technology/clustermap/index.html.

[7] Aduna Spectacle Manual. Technical Report. See http://aduna.biz/products/spectacle/docs/Aduna-Spectacle-2005.1-manual.pdf.

Developing SWS for e-Government

Leticia Gutierrez Villarias
Essex County Council
County Hall, Chelmsford
Essex, CM1 1LX
+44 (0) 1245 431990
Leticia.gutierrez@essexcc.gov.uk

Rob Davies
MDR Partners
2B Fitzgerald Road
London, SW14 8HA
+ 44 (0) 208 876 3121
rob.davies@mdrpartners.com

Mary Rowlatt
Essex County Council
County Hall, Chelmsford
Essex, CM1 1LX
+44 (0)1245 436524
maryr@essexcc.gov.uk

ABSTRACT

Categories and Subject Descriptors

C3 Special-purpose and application-based systems H4.2 Types of system

General Terms

Management, Design, Experimentation Human Factors, Standardization.

Keywords

DIP, ECC, e-Government, GIS, Ontology, Semantic Web Services.

1. INTRODUCTION

Essex County Council (ECC) - a local governmental authority within the UK - is the leader of the eGovernment case study within the European Commission-funded project DIP (Data, Information and Process Integration with Semantic Web Services (SWS)). Its aims are: to identify potential application areas for Semantic Web Services (SWS) as an infrastructure in real e-Governmental scenarios; and to improve the way in which multiple organisms (different departments within an organization, organizations and their customers, partners and suppliers) operate together to provide better services to citizens. Among the identified potential scenarios, two have been chosen for prototype implementation, in order to test the technology infrastructure for SWS created by DIP.

2. DIP Integrated Project

DIP's objective is to develop and extend Semantic Web and Web Service technologies in order to produce a new technology infrastructure for Semantic Web Services (SWS) - an environment in which different web services can discover and cooperate with each other automatically. DIP uses the Web Service Modeling Ontology (WSMO) as the overall framework for semantically enriching web services. WSMO is a formal ontology for describing the various aspects related to SWS following the Web Services Modeling Framework (WSMF), namely: Goals, Web Services, Ontologies and Mediators. WSMF is based on the following principle: *maximal de-coupling* and a *scalable mediation service* **Error! Reference source not found.**.

3. SWS in e-Government

Joined up services in e-Government almost always imply sharing scattered and heterogeneous data. SWS technology can help to integrate, mediate and reason between these datasets. SWS technology applied to the e-Government field promises to reduce risk and cost by: moving from "hard coding" services to reusable functionality; increase flexibility; enabling discovery of new or previously unknown services; aggregating services on the basis of user preferences; and providing better service to third-parties and customers.

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 11th - 14th June, 2006.

4. Change of Circumstances (CoC) Scenario

The first e-Government prototype within DIP is the "Change of Circumstances Scenario". This scenario is based on the *announcement of moving (change of address),* which is one of the twelve public services for citizens identified within the European Interoperability Framework for which the online sophistication is being benchmarked at national level.

Current service delivery to citizens is affected by a complicated inter-agency collaboration –a cross different tiers of Government (National, County, District or Borough) and external agencies – which makes it difficult to find the appropriate service to fulfil citizen's requirements. Relevant data has to be discovered and retrieved from a widespread array of heterogeneous data sources. SWS have the potential to be the technology which overcomes these difficulties and provides better services to the citizens.

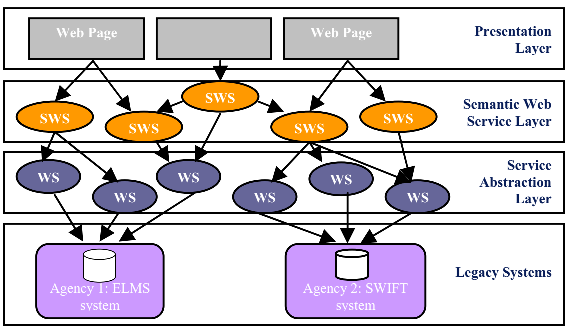
4.1 CoC scenario definition

The CoC scenario involves *"a single woman in part-time employment moving into a new house, in order to look after her disabled mother".* When this change occurs several agencies have to be notified (e.g.: County Social Services, District Housing Department, Department of Work and Pensions, etc.). The scenario makes use of SWS to seamlessly notify all the relevant agencies and provide to the citizen the benefits and/or services he is entitled to.

4.1.1 CoC architecture

The CoC prototype is a distributed system based on SWS

Figure 1 - CoC architecture



The CoC prototype accesses and automatically combines information from two real systems, namely the SWIFT and ELMS databases. The former stores information about citizens registered in ECC and their entitlement to services and benefits, while the latter stores information about equipment which is provided to citizens registered in Essex. The functionalities from these systems are accessed by means of Web Services, which are used as the basis for the SWS created. The Semantic Web Service Layer is WSMO compliant (consists of ontologies, mediators, web service descriptions and goals). This prototype is accessible from the user through a web GUI, depicted below in **Error! Reference source not found.**.

4.1.2 CoC Ontologies

The main ontologies are: an e-Government domain ontology that models a wide range of e-Government and community services and information, a specific task ontology about the SWIFT services and another about the ELMS services, which model how several agencies should be notified of a change of address or other circumstances of any person living in their area of competency in order to provide them with services and/or equipment.

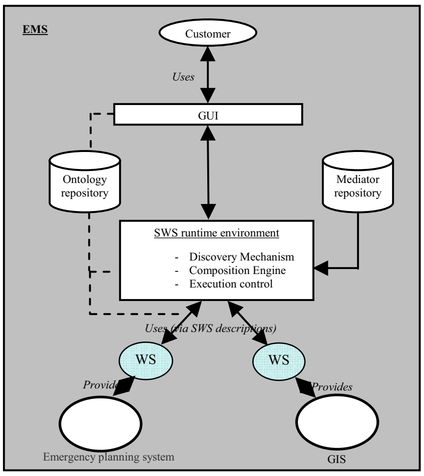
5. GIS Emergency Planning Prototype

The second prototype focuses on a Geographic Information environment. Many public organisations hold a large amount of Spatial-related Data (SRD) and manage a number of Geographical Information Systems (GIS). This wide distribution across the organisation causes duplication of data and lack of communication among SRD holders. Through implementation of a SWS-GIS system all the data would be made available through the web and its automated management would be possible (discovery, composition and invocation).

The focus is ECC Emergency Planning Department. More concretely, the scenario addresses a real past emergency situation around the Stansted airport area, namely the snowstorm on 31^{ST} January 2003, in order to ensure the availability of real data. The prototype is a decision support system (DSS), which assists the end user (emergency planners, police, ambulance, army, etc) in gathering information related to a certain type of event, quicker and more accurately.

5.1.1.1 Datasets

Figure 2 - GIS emergency planning architecture



Several emergency stakeholders are involved in order to create the most possible realistic scenario. Other agencies, together with ECC Emergency Planning Department, are collaborating in sharing their data, processes and expertise to be modeled in the ontologies.

The prototype emergency management system (EMS) is a distributed application which semantically selects and aggregates WS from the relevant agencies for a given emergency scenario – The emergency planning officer (EPO) accesses the application through a web-based GUI, utilising the Google Maps API, selecting and invoking a goal from the repository. This goal triggers the execution of one or more SWS. Several mediators can be invoked in order to cope with semantic mismatches between the services invoked. Several Ontologies support the semantic description of the remaining WSMO components.

5.1.2 Task Ontologies

A number of Ontologies have been developed to semantically support the SWS including an "emergency ontology" and ontologies for *"basic"*concepts, *"date and time"*, *"geographical concepts"*, and *"meteorology.*

The Ontologies have been created in WSML (Web Service Modeling Language), a language that that formalizes the Web Service Modeling Ontology (WSMO). They provide the semantic support to all the other WSMO components (WS, Goals and Mediators). Some of them describe the user's point of view (those describing the Goals) while some others describe the providers' point view of the world (those giving support to WS and Mediators). The inputs of the WS (XML in our particular scenario) are lifted to the ontology. After invoking a Goal, the results are lowered back into XML so the results can be displayed back to the user.

6. CONCLUSION

6.1 Foreseen Advantages

It is estimated that 30% of worldwide IT budgets is dedicated to Enterprise Application Integration (EAI) projects **Error! Reference source not found.**. By making use of SWS technology enterprises can exchange information and processes over the internet in a cost-effective way.

6.2 Overcome Difficulties

The foreseen advantages of SWS are completely new to the eGovernment sector and are chiefly visible only to academic/industrial research participants in the sector. Web Services - needed as the foundation for SWS - are only now beginning to be introduced as infrastructure (often experimental) in some government authorities. Despite this slow adoption, the awareness of need for semantic enrichment is increasing within the governmental sector and the DIP use case is contributing to this process.

Applying Semantic Web Technology in a Digital Library

Ian Thurlow
BT Group
Adastral Park, Ipswich
United Kingdom
+44 (0) 1473 609585
ian.thurlow@bt.com

Alistair Duke
BT Group
Adastral Park, Ipswich
United Kingdom
+44 (0) 1473 608192
alistair.duke@bt.com

John Davies
BT Group
Adastral Park, Ipswich
United Kingdom
+44 (0) 1473 609583
john.nj.davies@bt.com

ABSTRACT

One of the key aims of the SEKT project is to develop and exploit the semantic technologies that will underlie the next generation of knowledge management systems. A key element of the project is to evaluate and assess the impact of semantic web technology in case study settings. The overall aim of the case study, described here, is to investigate how the semantic web technologies being researched and developed in the project can enhance the functionality of a digital library.

Keywords

Semantic web, digital libraries, case study, SEKT.

1. INTRODUCTION

Semantically enabled technology is expected to bring a number of benefits to the users of corporate digital libraries. In particular, the technology will help people find relevant information more efficiently and more effectively, give better access to that information, and aid the sharing of knowledge within the user community of a digital library.

Section 2.1 gives some background information on the BT digital library. The key requirements for a semantically enhanced digital library are summarised in section 2.2. An overview of the BT digital library architecture is given in section 2.3. The prototype system is described in section 2.4. An outline of the software demonstration is given in section 3.

2. BT's DIGITAL LIBRARY

2.1 The BT Digital Library Today

BT subscribes to approximately 1000 on-line publications, giving end-users access to the full-text of over 900,000 scientific and business articles and papers. In addition, access is provided to over 4 million bibliographic records from the Inspec[1] and ABI[2] databases. A proprietary keyword-based search engine is used to search these information sources. A limited set of advanced search options are provided for the specialist or expert user, e.g. search by author's name, search by title or search by controlled indexing terms. Alternatively, users can browse the contents and abstracts of the library's journals. A prototype knowledge sharing application enables users to annotate web pages of interest. Other users can search these annotations.

2.2 Requirements

An extensive requirements capture exercise identified the following key requirements:

i) large amounts of relevant content are accessible on the Web - the content of the digital library should therefore be extended to include relevant web pages and RSS items,
ii) the bibliographic records from ABI and Inspec should be integrated with data sourced from the web using a common ontology,
iii) bibliographic metadata should be enhanced with richer metadata, e.g. identify named entities within a text.
iv) better search precision is required,
v) users should be able to annotate and share web pages with other registered users of the library,
vi) new applications should be supported by profiles that describe user interests, e.g. to give context to a user's search, or enable relevant information to be pushed to users.

2.3 The BT Digital Library Architecture

The BT digital library case study is based on the 5-layer SEKT[3] architecture [1].

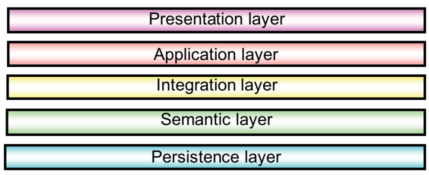


Figure 1. The BT digital library architecture.

The persistence layer comprises the internal sources of information, e.g. the Inspec and ABI bibliographic databases, and external sources of information, e.g. RSS items. The components that draw together relevant content for the digital library, e.g. the focused crawler, the components that populate the database, and the components that build profiles from an analysis of the log files are incorporated into this layer.

The semantic layer provides the components concerned with the creation, enhancement, maintenance, mediation, and querying of ontological information that is linked to the data stored in the persistence layer. Metadata associated with Inspec, ABI and RSS items is transformed into BT digital library ontology-specific

[1] http://www.iee.org/Publish/INSPEC/

[2] http://www.il.proquest.com/products/pt-product-ABI.shtml

[3] http://www.sekt-project.com/

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 11th - 14th June, 2006.

metadata. Named entities are extracted from texts by KIM[4], which employs GATE's[5] ontology-based information extraction module. The KAON2[6] system is used to store and reason over the resulting OWL[7]-based metadata.

The integration layer provides the infrastructure that enables the applications to be built from components in the semantic layer.

The principal case study applications, i.e. a semantic search and browse application, a semantic search agent, and a knowledge sharing application, are provided in the applications layer.

2.4 The BT Digital Library Prototype System

Squirrel, a tool for searching and browsing semantically annotated information, combines free text and semantic search with ontology-based browsing. Natural language summaries of ontological information are presented to the user. Search results are ranked, taking into account user profiles.

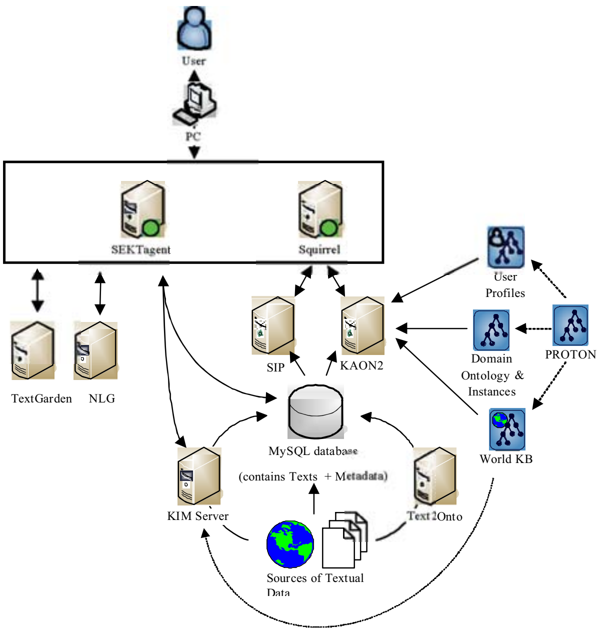


Figure 2. Squirrel and SEKTagent architecture.

SEKTagent, a semantic search agent facility, enables semantic queries to be specified, scheduled, and then invoked periodically over the digital library's pre-indexed documents. Relevant knowledge is delivered proactively to users. *Squidz*, a knowledge sharing application, enables a community of users to annotate Web pages of interest, share those annotations, and explore the interests and topics of other users. The tool builds upon current ideas of user tagging and community folksonomies and links user tags to a more formal ontology.

The applications are supported by a number of server side components that analyse textual documents and generate ontological information using the PROTON[8] ontology. A profile construction component, which is integrated with a web browser, enables profiles of users' interests to be constructed. A focused crawler enables relevant Web content to be added to the digital library. A classifier classifies Web content against topics in the BT digital library ontology (using vectors of co-occurring topics).

The BT digital library ontology is based on the PROTON ontology, which includes defines the top-level generic concepts required for semantic annotation, indexing and retrieval. This base ontology is extended with the additional classes and properties that are required to facilitate the SEKT-specific and case study-specific applications. The ontology has been populated with ABI and Inspec bibliographic data, along with Web content under a unified topic hierarchy. In addition, a world knowledge base (originally developed for the KIM platform) has been expressed in PROTON. This knowledge base comprises more than 200,000 entities, including around 36,000 locations, 140,000 companies and organisations, politicians, business leaders, technologists, etc.

The following components are used, either directly or indirectly, by the applications: PROTON, digital library extensions to PROTON, KIM, KAON2, semantic annotation, knowledge generation, knowledge repurposing, and the SEKT Integration Platform (SIP). Integration occurs at three levels: i) at the ontological level using a single overarching ontology on heterogeneous information sources, ii) at the component level using SIP (which allows SEKT technology components to be configured into data processing pipelines), and iii) at the application level, where applications are integrated into a portal.

3. SOFTWARE DEMONSTRATION

The demonstration is based on a typical usage scenario, i.e. a user views a set of *SEKTagent* results, configures a new search agent, navigates to the *Squirrel* tool, invokes some semantic queries and browses some meta-results, and finally uses *Squidz* to share a page with the digital library community.

4. ACKNOWLEDGMENTS

The work described in this paper was developed as part of the SEKT project, under EU funding (IST IP 2003-506826).

5. REFERENCES

[1] Semantic Web Technologies, Trends and Research in Ontology-based Systems, John Wiley & Sons, 2006.

---

4 http://www.ontotext.com/kim/

5 http://gate.ac.uk/

6 http://kaon2.semanticweb.org/

7 http://www.w3.org/TR/owl-features/

8 http://proton.semanticweb.org/

Integrating Semantic Web Services for Mobile Access

Anupriya Ankolekar, Pascal Hitzler, Holger Lewen, Daniel Oberle* and Rudi Studer
Institute AIFB
University of Karlsruhe
Karlsruhe, Germany
{ankolekar, hitzler, lewen, oberle, studer}@aifb.uni-karlsruhe.de

ABSTRACT
We present our work in integrating Semantic Web services for access via mobile devices. We have developed a system, the WebServiceAccessComponent, that transforms a user request for a service on a mobile device, to a Web service request and then selects a matching service from the existing Web services of the Deutsche Telekom, which provide navigational and weather information. In this poster, we present the requirements and design of the WebServiceAccessComponent.

1. INTRODUCTION
As mobile devices become increasingly widespread and as increasing numbers of companies expose their services as Web services, enabling flexible mobile access to distributed semantic Web resources for advanced personalisation and localisation features is a very relevant challenge. For example, a user can now potentially pose a open-domain question to her mobile digital assistant (MDA), which may even require knowledge about her context. The system can draw on her mobile context and use the Web as a knowledge base as well as a source for access to Web services to provide an answer immediately.

In this poster, we present our work in integrating existing Web services to provide answers to such questions posed by a user. Given a particular user question, e.g., "What will the weather be like tomorrow in Karlsruhe", the system must know that the answer can be delivered by a Web service, must know how to select/discover the corresponding Web service, must know if a composition of several Web service might be able to provide the result, and finally how to automatically invoke it.

Our system, the Semantic Mediator (see Figure 1) and specifically the WebServiceAccessComponent, has been developed in the context of the SmartWeb pro ject, which aims to demonstrate the feasibility of multimodal user interfaces to enable access to distributed semantic web resources and services on mobile devices. The SmartWeb system utilizes existing web services, including the T-Info (DTAG) web services [1] offered by Deutsche Telekom AG. The DTAG provides about 50 Web services providing dynamic information such as route planning, maps, weather information, GPS geocodes, locations of cinemas, playtimes of movies, events, points of interest (POI) and many more.

As can be seen in Figure 1, a query issued by the user on her MDA is transformed into an EMMA (Extended MultiModal Annotation markup language)[2] document, to represent the semantics of the query. This document is used by the Semantic Mediator to identify the Web services and knowledge resources that are required to answer the query. The Semantic Mediator then coordinates the access and invocation of these services and resources to return an answer to the user. Ontobroker is used as a storage and querying facility for Web service descriptions.

2. DESIGN OF THE WEBSERVICEACCESS-COMPONENT
The primary requirement for the WebServiceAccessComponent is that user queries in the form of EMMA documents must be dynamically matched to available Web services, to make the system robust and flexible. This also requires the selection and discovery of available Web services, given a particular user query. Using purely the XML Schema types specified in WSDL files, as Web services are typically described, it would be fairly difficult to automatically select and discover appropriate Web services. By semantically annotating available Web services, we describe the semantics of the inputs and outputs expected by the service. When the semantic annotations have been described through ontologies, a software program can reason about the services using the reasoning capabilities of the underlying logic of the ontology. We use an ontology inspired by OWL-S[3], SmartSUMO [2], but which address some of the shortcomings of OWL-S for our context.

To address these requirements, we essentially model existing DTAG Web services and attach semantic annotations to

*This work was done when the author was at Institute AIFB, University of Karlsruhe.

[1] `http://services.t-info.de/soap/index.jsp`
[2] `http://www.w3.org/TR/emma/`
[3] `http://www.daml.org/services/owl-s/1.1/`

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

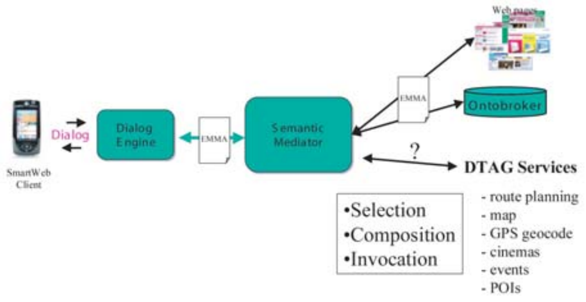


Figure 1: The Semantic Mediator in SmartWeb

them. Furthermore, we perform the selection and discovery of a matching Web service given a user question represented as an EMMA document. In the following, we enumerate the tasks involved for each of these and our design decisions for each of them.

1. Identification and representation of questions: For each DTAG service we have to identify possible natural language questions, based on the occurence of certain keywords, and to decide how they can be represented via EMMA and SmartSUMO, respectively. The actual descriptions are shown in the poster.

2. Extensions of SmartWeb ontologies: After deciding on the ontological representations of Web services, many additional concepts and relations were identified, which did not yet exist within the ontologies used in the SmartWeb project. The SmartWeb ontologies were therefore extended to enable a single ontology for the entire SmartWeb project.

3. Creation of EMMA documents: Given the identification and representation of various kinds of possible questions corresponding to all the DTAG Web services and the respective extensions of SmartSUMO, we then needed to specify how the EMMA documents for these queries should look like. This primarily involves configuring the natural language processing components of SmartWeb such that a particular EMMA document is created, given a particular question of the user.

4. Extensions of SmartSUMO for web service annotations: SmartSUMO itself needed to be extended to semantically annotate the DTAG Web services. This extension is separate from that mentioned in point 2, because these are related to describing the semantic web service characteristics themselves as opposed to describing the domain. Thus, we extended the SmartSUMO ontology to describe the concept of a Web service, to add concepts and relations for modelling semantic inputs and outputs of Web services, and to model the behaviour of a Web service. We discuss the last two extensions in the next two points.

5. Representation of web service inputs and outputs: Web service inputs and outputs are modelled as instances. A sequence of inputs is modelled via firstInput and nextInput relations, respectively. We have an ontoType relation to specify the semantics of an input by pointing to this instance. relevantSlot is required to match the correct Web service parameter if there are multiple parameters of the same type.

6. Representation of web service behaviour: We describe the behavioural aspects of Web services using the Ontology of Plans [1], an application of the Descriptions & Situations design pattern in DOLCE. A Plan consist of multiple Tasks, such as case, branching, synchronization, concurrency, or cycling task, etc., which are related through succession relations. We identify those tasks that are applicable for Web services in the SmartWeb domain and attach corresponding Plans to the DTAG Web services.

The annotation of all the DTAG Web services was performed manually. The selection and discovery of Web services was performed by making a simple assumption. We assumed that the focus of the EMMA document is the semantic output of the Web service, the (rdf:types of the) range of all direct relation to the focus documents are the semantic inputs. Thus, a Web service request can be easily created from an EMMA document. Given the Web service request, we then query the Ontobroker to find matching Web services by input and output type matching. Since we are currently dealing with only a few Web services, their invocation is hardcoded. We will be addressing the automated invocation and composition problems in future work.

3. CONCLUDING REMARKS

In this poster, we present the design of the WebServiceAccessComponent, which takes a user query for services or information and uses it to automatically discover and invoke appropriate existing Web services. This system is currently being used within the SmartWeb pro ject, which aims to provide multimodal access to distributed semantic Web resources via mobile devices. Of course, the context here is simpler than the usual Web services scenario in that there is no problem of ontology mediation, since the entire system uses a single ontology, the SmartSUMO ontology. We are also dealing with a relatively small set of 50 Web services, making the selection, composition and invocation problems considerably simpler. However, we believe that this system is still of value to the semantic Web services community as it is a real-world working system that demonstrates the feasibility and value of the semantic Web services scenario.

4. ACKNOWLEDGEMENTS

This document contains the work and advice of Robert Porzel (EML), Berenike Loos (EML), Hans-Peter Zorn (EML), Michael Sintek (DFKI), Malte Kiesel (DFKI), Daniel Sonntag (DFKI), Ralf Engel (DFKI), Massimo Romanelli (DFKI), Felix Burkhardt (Telekom), Jianshen Zhou (Telekom), Jin Liu (Telekom), Andreas Eberhart (HP invent). Thank you!

5. REFERENCES

[1] A. Gangemi, S. Borgo, C. Catenacci, and J. Lehmann. Task taxonomies for knowledge content. Technical report, Metokis deliverable d07, 2004.

[2] D. Oberle. Semantic management of middleware. In A. Sheth and R. Jain, editors, *The Semantic Web and Beyond*, number 1. Springer, New York, 2005.

Towards the Semantic Web in e-Tourism: Lack of Semantics or Lack of Content?

Martin Hepp
University of Innsbruck
Digital Enterprise Research Institute
Innsbruck, Austria
martin.hepp@deri.org

Katharina Siorpaes
University of Innsbruck
Digital Enterprise Research Institute
Innsbruck, Austria
katharina.siorpaes@deri.org

Daniel Bachlechner
University of Innsbruck
Digital Enterprise Research Institute
Innsbruck, Austria
daniel.bachlechner@deri.org

ABSTRACT
The Semantic Web aims at making the wealth of information available on the Web accessible to more precise search and automated information extraction and processing, based on a machine-readable representation of meaning in the form of ontologies. One common assumption is that the Semantic Web can be made a reality by gradually augmenting existing Web data by ontological annotations. In this paper, we describe the results of a quantitative analysis of Web content about Austrian accommodations and show that the data necessary to make the vision of the Semantic Web a reality is widely not available on the current Web, not even in a human-readable form. We discuss possible causes and argue that Semantic Web services technology will mitigate the problem, since business entities are more likely to expose functionality than data in competitive markets.

Categories and Subject Descriptors
K.4.4 **[Computers and Society]:** Electronic Commerce; I.2.4 **[Artificial Intelligence]:** Knowledge Representation Formalisms and Methods; J.1 **[Administrative Data Processing]:** Business; J.7 **[Computers in Other Systems]**

General Terms
Management, Human Factors, Documentation, Measurement

Keywords
Semantic Web, Semantic Web services, e-Tourism, Web, Annotation, Ontologies.

1. INTRODUCTION
One common assumption is that the Semantic Web can be achieved by gradually augmenting the existing data by annotations, and that the main problem of today's web is the "needle in the haystack" problem: everything is there, but we only have insufficient methods of finding and processing what's on the Web. In this paper, we show that, at least in the domain of eTourism, this assumption is inappropriate, because the available Web resources do not contain sufficient information, even from a perspective of a human user. As a consequence, even a perfect annotation of existing Web content would not allow the vision of the Semantic Web to become a reality in the domain of e-Tourism. We analyzed websites operated under the direct control of the accommodation management and also such maintained inside tourism portals. At this point, we abstract from the task of annotation itself, i.e. to which degree the process of adding machine-readable meaning to existing content. In section 2, we describe the methodology for our analysis. Section 3 summarizes our data and highlights core findings. In section 4, we discuss the implications of our findings for Semantic Web research. Section 5 concludes the paper.

Demos and Posters of the *3rd European Semantic Web Conference (ESWC2006),* Budva, Montenegro, June 11-14, 2006.

2. METHOD
First, we identified information categories that are relevant for consumers looking for travel accommodation. For this, we reused a survey by the Austrian Chamber of Commerce [1], which describes the information needs of consumers. We added the category "availability", which is not listed explicitly in the survey, since it is a core information need in typical travel-related scenarios. Then, we created an ordinal scale for the amount of information per category, ranging from 0 (no information) to 5 (comprehensive coverage of all aspects). Second, we obtained the official directory of all legal accommodations located in the state of Tyrol (n=4,665). Third, we took a random sample (n=100) of the listed accommodations, and for each entry in this sample, searched the Internet for an official Web page. If we could not find a Web resource or if we had doubts about the identity, we called the owner or operator of the accommodation for clarification. Fourth, we checked the leading Austrian tourism portal Tiscover (http://www.tiscover.at) for entries covering the very same sample. Fifth, we manually analyzed the content of both the respective vendor-operated Web resources and the Tiscover entries, and graded the amount of available information using the predefined ordinal scale. Sixth, we aggregated the results and determined the amount of Web resources and portal entries that provide at least a "sufficient" amount of information in the respective category according to the grading scheme. Sufficient was defined in the sense that all information is given that an average consumer needs in order to determine his or her perceived utility of an available accommodation, i.e. to make a reservation decision.

3. RESULTS
In this section, we present the results of the survey and highlight significant observations. We clearly distinguish between vendor-provided data and portal data, since the portal Tiscover, which was part of our analysis, is a managed Web site and does not grant external access to the full internal database. The potential contribution of Semantic Web technology for tourism *portals* is likely smaller as compared to individual Web sites, since portals put a lot of effort into resolving inconsistent data representation

and thus mitigate the problem of data heterogeneity and other obstacles that Semantic technology promises to overcome.

3.1 Representation of Accommodations in the WWW

Out of the 100 accommodations in the sample, 60 maintain a Web site individually for this accommodation, either operated by the hotel owner or managed by a service provider. Additionally, all of these 60 are members of the Austrian tourism portal Tiscover. 33 are only represented in the Tiscover tourism portal. 5 % cannot be found at all on the Web but their existence could be verified by phone, and 2 % do either not exist any longer or could not be found at all.

3.2 Coverage of Information Categories

In this section, the percentage of Web sites that contain at least a sufficient amount of information in the various categories is visualized in Figure 1. As detailed above, sufficient means a rating of 3 points or more in the grading scheme.

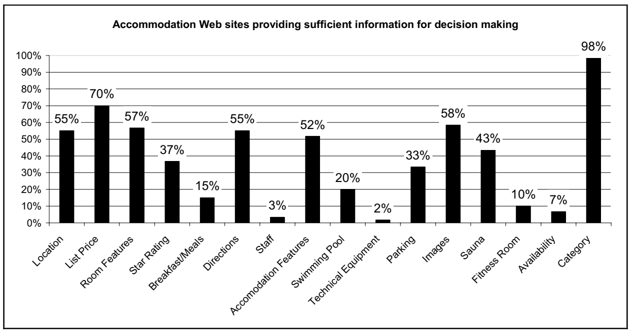


Figure 1. Percentage of specific Web sites with sufficient information for decision making.

3.3 Availability of Information in Tourism Portals

In this section, we summarize the coverage of consumers' information needs among the 93 hotels that are listed in the Tiscover portal. We were surprised by the following observations:

1. Only 27 % of those hotels from our sample that are listed in Tiscover give price information (whereas 70 % of the hotel Web pages contain at least a list price). 2. Only a third of those hotels have sufficient room feature descriptions in Tiscover, while more than half (57 %) of the vendor-operated Web pages contain such detail. 3. 20 % of all listings in Tiscover give sufficient information about technical equipment, while only 2% of the Web pages contain such detail. 4. Three times as many hotels give current availability information on Tiscover (22 %) as compared to vendor operated Web sites (7 %). Still, the biggest part of all Tyrolean hotels does not provide current availability information anywhere on the Web.

4. DISCUSSION

In our sample only 60% of the accommodations maintain a website, while 93% are represented in a tourism portal. Thus, it is very likely that a portal membership is more feasible than a self-maintained website. This is an important quality when assessing the potential of Semantic Web technologies in this sector. From the results we can see that the self-operated websites of accommodations lack information. Only 7 % offer room availability information, which is the most important fact when searching for a suitable offer. The remaining 93 % of accommodation Web sites require a user to either call or communicate by e-mail with the provider in order to get availability information. This is a serious obstacle for making the Semantic Web a reality in the E-tourism domain. The situation inside the tourism portal Tiscover is remarkably better, but in several ways still surprisingly insufficient. For eight out of ten hotels, no current availability data is available, and for 73 % of the hotels not even a list price can be retrieved. The predominance of tourism portals is a challenge for the Semantic Web, since the internal databases are not accessible, and the discovery and matchmaking of consumer request and available supply is hidden inside the portal. As a consequence, the Semantic Web cannot be made a reality in the sector we analyzed by annotating information on Web pages ("data-centric Semantic Web"). Rather, turning the Web into the Semantic Web requires annotating exposed functionality, i.e. services.

5. CONCLUSIONS

We have presented evidence that the current quantity of information available on the Web would not allow the Semantic Web to become a reality in the domain of accommodations in Tyrol. Our analysis has shown that dedicated Web sites do not contain sufficient information that would allow potential guests to make a reservation decision without additional e-mail or phone communication. Similarly, information in managed tourism portals is insufficient, while, additionally, those portals keep control over all functions and the data inside their databases, which are not exposed on the Web. Since we have no reason to assume that the encapsulation of information inside systems will decrease, we can assume that the Semantic Web will only become a reality, in the domain of e-Tourism, if it includes the annotation of functionality and not just published information. In short, the vision of the Semantic Web will not become a reality without Semantic Web services tech-nology, e.g. WSMO or OWL-S.

6. REFERENCES

[1] S. Dolnicar and T. Otter. Marktforschung für die oesterreichische Hotelklassifizierung (Market research for the Austrian hotel classification schema; in German). Vienna, Austrian Chamber of Commerce, 2001.

[2] Fodor, O. and H. Werthner: Harmonise: A Step Toward an Interoperable E-Tourism Marketplace. International Journal of Electronic Commerce 9 (2), 11-, 2004.

[3] Dogac, A., Y. Kabak, G. Laleci, S. Sinir, A. Yildiz, S. Kirbas and Y. Gurcan: Semantically Enriched Web Services for the Travel Industry. SIGMOD Record 33 (3), 21-27, 2004

Litteratus Calculus: a manifesto for a demographic way to build a sustainable Semantic Web

Jean Rohmer
Centre des Nouvelles Technologies
d'Analyse d'Information (CENTAI)
Thales Communications France
jean.rohmer@fr.thalesgroup.com

ABSTRACT

Litteratus Calculus is a proposal to help Semantic Web enjoy the good properties which made the success of the original Web, i.e. natural language, navigation, simple search, freedom of formats, without sacrificing the structural and normative qualities of semantics. This poster outlines the underlying formalism of Litteratus Calculus, explains how it relates to Semantic Web standards, and describes some promising experiments. It is an invitation to a shared reflection on these points.

KEYWORDS: semantic and natural language, semantic writing, semantic web evolution

1. IN SEARCH OF ANOTHER BOOSTER FOR THE SEMANTIC WEB

In parallel with a growing interest from the scientific IT community, eagerness to see the Semantic Web (SW in short) more widely used is a constant since its inception. See for instance the search of its "killer app" [1]. This continuous expectation contrasts with the speed at which the original Web spread out. In some sense, with the Web, Sir Tim Berners-Lee gave us the solution, when for the Semantic Web he gave us the challenge. Which may reveal equally fruitful if we accept to lucidly address it.

We feel that two dangers threat the SW. The first one is to be seen as pure Software Engineering: "another way to develop applications with data models, programming languages, programmers". The second danger is to consider that formal logic should be the preferred representation and computation mechanism for the SW. Another idea to boost the SW is to automatically convert documents into RDF statements. Unfortunately, Software Engineering, Formal Logic and Natural Language Understanding are long, difficult ventures, which started 40 years ago or much before, and which today cannot deliver instant miracle solutions. (See for instance [8], a recent deep reflection of the Software Engineering Community about its future).

Challenging this semantic *morosité,* new approaches like Semantic Desktops[4] and Semantic Wikis [3], have in common to target very large communities of SW writers. This is, in our point of view, the key point, the ***demographic point***: if we limit ourselves to the thousands of people who speak formal logic, or to the millions of software engineers, the Semantic Web will become the *Semantic Wait …*

Litteratus Calculus uses as few software engineering, formal logic and computational linguistics as possible so as to dramatically increase the number of contributors to the SW.

Note 1. The slow growth of the SW may originate from its aim for a better understanding between machines, which got the main visibility at the detriment of more people-oriented features [2].

Note 2. As soon as initiatives like Semantic Desktops, Semantic Wikis and hopefully Litteratus Calculus will yield a new *semantic compost,* the above-mentioned disciplines might indeed become more instrumental

2. LITTERATUS CALCULUS

In Litteratus Calculus (LC in short) *litteratus* stands for *people* and *calculus* for *machines*. LC can be seen as a generalization of Semantic Networks (SN in short), which, as shown by [7] are one of the preferred formalisms to represent formal knowledge since the Ancient Greeks. Ubiquitous in Artificial Intelligence, they were a natural choice when the idea of the SW arose. However they seem too close to machines and too far from people, since in one hand Software Engineering and Formal Logic so easily put a grasp at them, and since in another hand it is so difficult to automatically translate text to them.

<u>Litteratus Calculus first step</u> is to replace the SN notion of triple (e.g. Subject / Verb / Object) by the notion of *minimal autonomous sentence.*

A minimal autonomous sentence is a sentence in any human readable language which is *atomic*, i.e cannot be replaced by one or more shorter sentences, and *autonomous* in the sense it is self-sufficient to be understood alone by some community of people. We call such a sentence an *inferon.*

RDF statements *are inferons*, Logic Clauses *are inferons*, Natural Language sentences *may be* inferons. Examples: (the first sentence of this paragraph)

"In the expression Litteratus Calculus (LC in short) *litteratus* stands for *people* and *calculus* for *machines." <u>is not an inferon: not minimal</u>*

"(LC in short)" ***<u>is not an inferon: not autonomous</u>***

"In the expression Litteratus Calculus *litteratus* stands for *people"* <u>is an inferon: minimal and autonomous</u>

<u>Litteratus Calculus second step</u> is to consider, given two inferons, the *set* of their common words, called *interlogos.* Example:

Inferon 1: "In the expression Litteratus Calculus *litteratus* stands for *people* "

and Inferon 2: "Litteratus Calculus is designed for people"

have in common the *interlogos* "Litteratus Calculus for people"

Finally, we call ***argos*** a set of ***inferons*** with their ***interlogos. An argos is a bipartite sub-graph of inferons and interlogos***.

3. LITTERATUS CALCULUS: MORE CONTRIBUTORS, AND MORE RESPONSIBLE ONES

The Litteratus Calculus project can now be made more explicit:

- Let people write inferons *–and only inferons-*
- Build automatically the resulting *argos* and *interlogos*
- Provide tools to make navigation, computations, inferences on the *argos* network
- Provide tools based on analogy and emergence to assist people writing new *inferons*

Key points which will be illustrated in the Poster:

Everything is represented with *inferons:* facts, ontologies, rules, queries …: there is an unique *"semantic soup of inferons"*

The main logic / computational operation is analogy. For instance, using a rule or query is making an analogy between the *argos* representing the rule or query and *argos* of basic facts (everybody understands analogy, which is not the case for formal logic)

Two basic operations in LC permit to compute resemblance and differences between *inferons, interlogos, argos*, based upon graph topology. For instance the resemblance between two *interlogos* is a set of *argos*. (Consider: "Jack and John work in companies which both have customers which receive grants from projects sponsored by European Union")

Inferons writing should be an altruist activity: when typing a new *inferon*, the user can see how it relates to existing ones -through *argos*. He can see for instance how it closes circuits, helping to fire rules or answer queries. If the writer considers that his new *inferon* is not well connected, he may decide to add *supplementary inferons* to fill the gap (exactly as we proceed during a face-to-face conversation). This way, useful ontologies will be built in a need-driven process. LC encourages *alterity* (alter IT!). The more *responsive* is the system, the more *responsible* becomes the user. LC aims at replacing Software Engineering by a conscious discipline and altruism from users.

4. PRACTICAL STEPS TOWARDS LITTERATUS CALCULUS

All the expected LC properties remain valid if we restrict ourselves to simple inferons (RDF, SVO, triples …). And in fact we experienced that it is extremely fruitful to explore two tracks in parallel : one "low" track with simple Semantic Networks, one "high" track with true natural language *inferons*.

As reported in [6], we develop and use since 1993 a simple Semantic Networks Editor, IDELIANCE, which has been intensively used by individuals and groups to write and share knowledge (users include French Military Intelligence, L'Oreal, Air Liquide, Merck Pharmaceutical Labs, Thales). With IDELIANCE, users of various professional profiles directly create shared semantic networks, after some hours or days –age depending- of training. We call this low track Litteratus Calculus "A", and the high track Litteratus Calculus "B".

LC "A" is of course easier to manipulate inside a machine. This generates many ideas of algorithms like: computing all *argos* between two subjects, all circular *argos* visiting a given set of subjects, and filter *argos* according to the nature of subjects and verbs they are made of. We also developed mechanisms of suggestions when writing new *inferons*: given the current environment of a subject in the graph, users are suggested new statements by analogy with similar graph configurations.

Once proved in LC "A" , these features can then be taken as objectives to be transposed in LC "B", no longer in terms of subjects and relations, but in the more complex, less formal lattice of *inferons, interlogos and argos*. These transpositions from LC "A" to LC "B" often invite us to state the problem in more general terms, leading to more general features, which, in return, give new specifications for LC "A".

One can ask the question: why not concentrate on LC "A" and improve it ? The answer refers to our *demographic point*: we noticed that at most 1 (one) per cent of people –among a normal business population- spontaneously adhere to LC "A" –rather than remaining in the traditional textual / document mode.

Our bet is, with LC "B", to raise this percentage to about 10%. (it is important to realize that writing *inferons* ***is not*** writing plain text as usual. Even with much less constraints than with LC "A", it harnesses people's reflection –not a bad point in other respects)

One of the initial possible outcomes of Litteratus Calculus is to promote new ways of scientific publishing, as anticipated and proposed in [5].

5. REFERENCES

[1] Alani H, Kalfoglou Y, O'Hara K, Shatbolt N: Towards a Killer App for the Semantic Web, ISWC 2005 Galway , Ireland

[2] Berners-Lee Tim, Hendler Jim, Lassila Ora, The Semantic Web, Scientific American May 2001

[3] Campanini S., Castagne P., Tazzoli R, Paltypus Wiki, 3rd International Semantic Web Conference (ISWC 2004) Hiroshima,Japan

[4] Decker S, Park J., Quan D., Sauermann L. First Semantic Desktop Workshop 2005 Galway Ireland

http://www.semanticdesktop.org/xwiki/bin/SemanticDesktopWS2005/

[5] De Waard A., Oren E., Haller H., Völkel M., Mika P. Schwabe D. The ABCDEF Format For Conference Proceedings

http://wiki.ontoworld.org/index.php/ABCDEF

[6] Rohmer Jean, Lessons for the future of Semantic Desktops learnt from 10 years of experience with the IDELIANCE Semantic Networks Manager ISWC 2005 Galway , Ireland (November 2005)

[7] Sowa John F. Semantic Networks

http://www.jfsowa.com/pubs/semnet.htm

[8] Wirsing M. Editor, Software Intensive Systems, Draft Report on ERCIM "Beyond the Horizon Thematic Group 6" (January 2006)"

ONTHOLOGY - An Ontology Metadata Repository

Jens Hartmann
Institute AIFB, University of Karlsruhe, Germany
hartmann@aifb.uni-karlsruhe.de

Abstract

Most ontologies today exist in pure form without any additional information, e.g. authorship or domain specific information. The proposed Ontology Metadata Vocabulary (OMV) aims to establish a standard which enables users from academia and industry to identify, find and apply – basically meaning to reuse – ontologies effectively and efficiently. Our demo/poster contribution consists of the presentation of the up-and-running metadata portal ONTHOLOGY ("anthology of ontologies") which implements the proposed OMV to support users in accessing and reusing of ontologies. OMV is available for download at `http://omv.ontoware.org/`, ONTHOLOGY is available at `http://www.onthology.org/`.

1. Introduction

Ontologies are commonly used for a shared means of communication between computers and between humans and computers. To reach this aim, ontologies should be represented, described, exchanged, shared and accessed based on open standards such as the W3C standardized web ontology language OWL. However most ontologies today exist in a pure form without any additional information about authorship, domain of interest and other meta data about ontologies. Ssearching and identifying existing ontologies which are potentially reusable because they e.g. are applied in similar domains, used within similar applications or who have similar properties is a rather hard and tedious task.

We argue that metadata in the sense of machine processable information for the Web[1] helps to improve accessibility and reuse ontologies. Further, it can provide other useful resource information to support maintenance. Thus we claim that metadata not only help when applied (or, attached) to documents, but also to ontologies themselves.

As a consequence, ontologies which are annotated by metadata require an appropriate technology infrastructure as well. This includes tools and metadata repositories which comply to the ontology metadata standard and which provide the required functionalities to support reuse of ontologies. Such tools and repositories typically should support the engineering process, maintenance and distribution of ontologies.

In this paper we present the up-and-running portal infrastructure ONTHOLOGY as reference implementation which shows the benefit of applying such standard in a centralized scenario. The main functionality of the portal is to store, manage and making accessible ontology meta data for large user communities.

2. Ontology Metadata Vocabulary (OMV)

The presented metadata portal stores information according the metadata vocabulary OMV which has been proposed as metadata standard in the EU IST thematic network of excellence Knowledge Web[2].

OMV distinguishes between an **ontology conceptualisation** and an **ontology implementation**. This separation is based on following observation: any existing ontology (implementation) has some kind of *core idea* (conceptualisation) behind. From an ontology engineering perspective, a person initially develops such a *core idea* of what should be modeled in his mind. Further, this initial conceptualisation might be discussed with other persons and after all, an ontology will be *realised* using an ontology editor and stored in a specific format. Over time there might be created several *realisations* of this initial *cconceptualisation* in many different formats, e.g. in RDF(S)[3] or OWL[4].

The distinction between an ontology conceptualisation and ontology implementation leads to an efficient mechanism, e.g. for tracking several versions and evolvements of ontologies as well as for different representations of one knowledge model (conceptualisation) in different ontology languages. Such an *ontology conceptualisation* can be seen as representation of the conceptual model behind an ontology.

Besides these two main classes, additional classes are re-

[1] `http://www.w3.org/Metadata/`

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

[3] `http://www.w3.org/RDF/`

[4] `http://www.w3.org/TR/owl-features/`

quired to represent useful information about ontologies by such vocabulary. Therefore OMV provides further classes and properties representing *environmental information* and *relations*, e.g. such as *persons, engineering tools* or even *license models*. The complete metadata ontology is illustrated in [1].

3. ONTHOLOGY — Ontology Metadata Repository

As the importance of metadata increases with the number of existing ontologies, the demand for a supporting technologies like storage and access techniques becomes important as well. We present the conceptual design of a centralised ontology metadata repository and its implementation, so-called ONTHOLOGY standing for "anthology of ontologies".

Centralised systems allow to reflect long-term community processes in which some ontologies become well accepted for a domain or community and others become less important. Such well accepted ontologies and in particular their metadata need to be stored in a central metadata portal which can be accessed easily by a large number of users whereby the management procedures are well defined. A main goal of a centralised metadata portal is to act as large evidence storage of metadata resp. their related ontologies to facilitate access, reuse and sharing as required for the Semantic Web.

We identified several different user roles for ONTHOLOGY: The *visitor* is an anonymous user, he is allowed to browse the public content of the portal. A Visitor can become a *user* by completing an application form on the website. In order to avoid unnecessary administrative work, a user is added automatically to the membership database. Users can customize their portal, e.g. the content of their start-page or their bookmarks. If a user wants to submit metadata to the portal, this submission has to be reviewed before it is published. ONTHOLOGY establishs a *review process* in order to ensure a certain level of quality. *Reviewers* check the new submissions before it is published. The *technical administrator* is responsible for any other task mainly the maintenance of the portal.

Functionalities of ONTHOLOGY can be separated into two groups based on the usage. Indeed, *basic functionalities* which are provided to every user who accesses the portal and *sophisticated functionalities* for reviewers and administrators. The main operations a user can perform on the repository are (i) *Search*, (ii) *Submit* and (iii) *Export*.

The search and export can be performed by any visitor without being registered to the repository. Since providing new metadata is based on a certain community confidence, a visitor has to register at the portal to be become a registered user.

A metadata portal mainly consists of *a large data repository* in which metadata can be stored. Exemplary, Sesame[5] or KAON[6] can be used as back-end metadata repository. *Access* and in particular the *managament* of the repository must be guaranteed, too. Therefore, ONTHOLOGY is based on SEAL, the AIFB conceptual architecture for building SEmantic portALs. In SEAL ontologies are key elements for managing community web sites and web portals. They support queries to multiple sources, but beyond that also intensive use of the schema information itself to allow for automatic generation of navigational views such as navigation hierarchies that appear as `has- part`-trees or `has- subtopic` trees in the ontology. In addition to that mixed ontology and content-based presentation is supported. Further information can be found at [2].

In addition to the central storage and maintenance, ONTHOLOGY cooperates with the decentralised system Oyster[7] which stores and retrieves metadata in a P2P manner. The benefit of connecting both systems lies mainly in the simple reuse of existing ontology metadata information from such networks of users who are willing to share them. Whereas the portal is expected to contain data which matures according to quality insurance procedures over time, the ad-hoc P2P network enables quick and easy distribution of data without much control. In combination, both systems ensure efficient and effective ontology metadata management for various use cases.

4. Conclusion

To conclude, reusing existing ontologies is a key issue for sharing knowledge on the Semantic Web. Our contribution aims at facilitating access and reuse of ontologies which are previously unknown for ontology developers and users through the ONTHOLOGY metadata portal. As metadata standard we use the Ontology Metadata Vocabulary (OMV). Next steps include the standardization of OMV on a wider scope and the development of further extensions to ONTHOLOGY, in particular the linking of ONTHOLOGY with Oyster requires additional efforts.

5. References

[1] J. Hartmann and R. Palma. OMV - Ontology Metadata Vocabulary for the Semantic Web, 2005. v. 1.0, available at `http://ontoware.org/projects/omv/`.

[2] J. Hartmann and Y. Sure. An infrastructure for scalable, reliable semantic portals. *IEEE Intelligent Systems*, 19(3):58–65, May/June 2004.

[5] `http://www.openrdf.org/`

[6] `http://kaon.semanticweb.org/`

[7] `http://ontoware.org/projects/oyster`