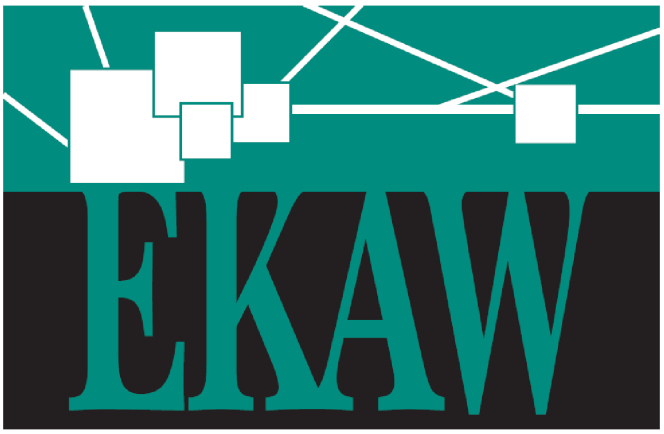


2006

POSTER AND DEMO PROCEEDINGS

of the 15th International Conference on Knowledge Engineering and Knowledge Management

Managing Knowledge in a World of Networks

Helena Sofia Pinto

Martin Labský (Eds.)

Poděbrady, Czech Republic, October 2006

EKAW 2006

Poster and demo proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management

http://ekaw.vse.cz

Poděbrady, 2nd - 6th October 2006

Supported by:

Department of Information and Knowledge Engineering

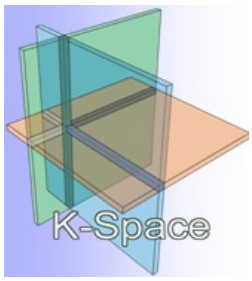
University of Economics, Prague

Czech Republic

K-Space: Knowledge Space of Semantic Inference for automatic annotation and retrieval of Multimedia Content

NeOn: Lifecycle Support for Networked Ontologies

X-MEDIA: Large-scale knowledge sharing and reuse across media



ISBN 80-86742-15-6

Preface

EKAW is concerned with all aspects of eliciting, acquiring, modelling, managing and exploiting knowledge, and the role of these aspects in the construction of knowledge-intensive systems and services. Topics of interest include but are not limited to:

A) ONTOLOGIES AND THE SEMANTIC WEB
- Languages for ontologies and the semantic web (SW)
- Methods and tools for collaborative building, evolution and evaluation of ontologies
- Knowledge modelling, knowledge management and knowledge evolution on the SW
- Ontology learning
- Ontology reengineering, reuse, merging, alignment, integration and certification
- Ontologies for information sharing and intelligent information integration
- Ontology-based annotation
- Multimedia technologies and the semantic web
- Semantic Wikis, Semantic Bloggs

B) SEMANTICS FOR GRID AND WEB SERVICES
- Semantic web services: Theory, Tools and Applications
- Semantic grid services: Theory, infrastructure and Applications
- Knowledge Services for e-Science
- Problem solving methods and semantic web services
- Peer to Peer communication between semantic systems
- Brokering systems
- Ontologies and agents
- Semantic Portals

C) KNOWLEDGE AND SOCIAL NETWORKS
- Social and human factors dimensions of knowledge management
- Knowledge and social network analysis & modelling
- Knowledge in trust networks
- Social Tagging and Folksonomies

D) KNOWLEDGE MANAGEMENT
- Methodologies and tools for Knowledge Management
- Methodologies and tools for corporate memory construction, evaluation and evolution
- Knowledge modelling and enterprise modelling
- Human language technologies and Knowledge Management

E) KNOWLEDGE ACQUISITION and MODELLING
- Advanced Knowledge modelling languages and tools
- Knowledge capture through machine learning and knowledge discovery in databases
- Specific knowledge modelling issues for CBR systems, coop. KBS, training applications
- Knowledge Acquisition from texts and WWW
- Evaluation of methods, techniques and tools for Knowledge Acquisition
- Knowledge engineering and software engineering
- Uncertainty and vagueness aspects of knowledge modelling

Besides the scientific track, EKAW asked for poster and demonstration contributions that were presented in a special session during the conference. For the poster & demo session we were looking for contributions whose nature make them less suited for submission to the official paper track. In particular, we asked for contributions of the following kind:

Late-breaking and Speculative Results: Significant and original ideas with promising approaches to resolve open problems in 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 Knowledge Engineering and Knowledge Management 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 KE&KM.

We received 29 submissions covering all aspects of Knowledge Engineering and Knowledge Management research which involved researchers from 15 different countries, in three different continents: America, Australia and Europe. All submissions were reviewed regarding their suitability for the poster and demo session by a dedicated program committee. We were able to accept 21 of these submissions for presentation at the poster & demo session to be held on Wednesday, the 4th of October 2006. We would like to thank all those who supported the poster and demonstration session of EKAW 2006. Special thanks go to the members of the program committee for helping us to select interesting contributions.

Finally, we hope the poster & demos session provides its attendees lively, fruitful and interesting discussions on all aspects of KE&KM research. For those who are not able to attend the session we hope these proceedings can be useful as a showcase to the last developments in the KE&KM area.

Poster Chair:
Helena Sofia Pinto, Technical University of Lisbon (PT)

Demo Chair:
Martin Labsky, University of Economics, Prague (CZ)

Poster & demo Program Committee

Stuart Aitken	Edinburgh University (UK)
Harith Alani	Southampton University (UK)
Philipp Cimiano	University of Karlsruhe (DE)
Oscar Corcho	Manchester University (UK)
Mariano Fernández Lopez	Fundación Universitaria San Pablo CEU (ES)
Marko Grobelnik	Jozef Stefan Institute (SL)
Siegfried Handschuh	DERI (IR)
Andreas Hotto	Kassel University (DE)
Michael Klein	Vrie University (NL)
Andreia Malucelli	Pontifical Catholic University of Paraná (BR)
Peter Mika	Vrie University (NL)
Alun Preece	University of Aberdeen (UK)
Marta Sabou	Open University (UK)
Christoph Schmitz	Kassel University (DE)
Sergej Sizov	Koblenz-Landau University (DE)
York Sure	University of Karlsruhe (DE)
Maria Varas-Verga	Open University (UK)

Table of Contents

New approach for document automatic annotation

Lylia Abrouk
LIRMM
161 rue Ada
34392 Montpellier Cedex 5
abrouk@lirmm.fr

ABSTRACT
Being involved in the euro-mediterranean water information system (EMWIS) which goal is to diffuse and facilitate the access to the information related to the water sector. We focus our interest on the description of this information in order to ease the exchange and the discovery of documents. This paper describes an approach to automatically annotate documents for the EMWIS. This approach is based on the cited references; this is done in order to propagate their annotations on the target document. To achieve this, we use co-citations method and clustering algorithms. The evaluation of this work is done thought the System we have developed: RAS (Reference Annotation System).

General Terms
Annotation, clutering, ontology, emwis

1. INTRODUCTION
For many fields, the Web and its technologies became the greatest source of current information. But the specificity of such sources of information makes them not easily exploitable and their constant evolution makes complex the search for information. The principal reason is as follows: the documents are dispersed, heterogeneous and often are not structured. It is necessary to propose methods and tools making it possible to share, manipulate and search in such documents.
The semantic annotation using ontologies is currently the most relevant method and most promising to mitigate the problems of volatility and heterogeneity of the documents on the Web. The annotations make it possible to associate information additional to the documents, to specify certain parts of this one and finally to share them within the framework of a working group. This annotation is thus very useful to refine the answers to the requests of the users. Nevertheless, the semantic annotation raises two principal problems:

- resource annotation: in fact, within a large system, it is not feasible to assume that the content of all the resources can be described manually by experts. Our goal is then to provide a mean to assist experts and content managers to annotate the resources by suggesting automatically some annotations after analysing the citation links of already existing resources;
- global ontology enhancement: an ontology is a structured whole of concepts. The concepts are organized in a graph whose relations can be: *(i).* semantic relations, *(ii).* composition and heritage relations. We should find a technique to add new concepts and relationships within the global ontology and update automatically the existing annotation of resources.

This paper targets only the first part of the work and presents means in order to annotate automatically a large set of resources using the citation links that structurally exist among resources. This approach annotates documents without knowing their content basing on references.

2. ANNOTATION APPROACH
To implement this solution, one has to answer the following questions: *(i)* what citations should be taken into account? In fact, not all the citations in a document are meaningful to determine the theme of the document; *(ii)* How to annotate the document? *(iii)* and finally, how to merge annotations that come from the selected documents.

Do add new document d, we proceed like this:

1. recover the whole of the documents cited by d in a set Ref_d;
2. thematic group the documents of the Ref_d set in order to determine the most relevant group of themes and to thus avoid the references non relevant but present in Ref_d;
3. import the annotations of the documents cited by d;
4. select among the annotations been essential most relevant to propose them as annotation of document d.

2.1 Thematic group
When an author cites another document, this is done to indicate that the cited document contains some information which is relevant to the context of the citation. However,

we can also find citations that contribute to a small part of the document and do not necessarily determine the general theme of the whole document. Consequently, we have to consider only citations that contribute to determine the thematic of the source document. The co-citation method has been proven to be a good measure to determine the similarity on theme among documents. In fact, when documents are often cited together by different documents, we can assume that they target the same subject. We define a measure $S_{(i,j)}$:

$$S_{i,j} = \frac{1}{{C_{(i,j)}}^2} \quad (1)$$

$C_{(i,j)}$ is the co-citation frequency or the number of time that i and j are cited together; The equation 1 takes into account simply the co-citation index between two documents in order to determine their thematic proximity. Thus, more the documents are cited together, more the distance $S_{(i,j)}$ will be close to zero.

When the references of the document d are recovered, we build the graph of citation GC_d:

$$GC_d =< Ref_d, Ref_d \times Ref_d \times [0,1] > \quad (2)$$

As described in the equation 3, the graph of citation is a complete graph where the nodes represent the documents cited in d, and a link between two documents i and j is a valuate link with the function of distance $S_{(i,j)}$ presented in the equation 1. The representation of this graph can also be seen like a matrix, called matrix of citation, MC, defined as follows:

$$MC_d : |Ref_d| \times |Ref_d|$$
$$\forall i,j \in Ref_d,\ MC_d(i,j) = \begin{cases} S_{(i,j)} & \text{si} \quad i \neq j \\ 0 & \text{else} \end{cases} \quad (3)$$

From this matrix, we can search the groups (clusters) of close documents. For this reason we use an algorithm for clustering fuzzy c-means' [2] which uses the fuzzy set theory. To use this matrix as input for the algorithm ' fuzzy c-means' it should be taken into account that the valuations of the links define a mathematical distance. Indeed, as the documents are independent and than the calculation of $S_{(i,j)}$ takes into account only the index of cocitation of these documents, the specification of the mathematical distance can not be satisfied.

Generally, we can have a cumulated distance on a path connecting two documents which is lower than the direct distance between two documents. In this case, the graph of citation does not present a mathematical distance and the use of the algorithm ' fuzzy c-means' will not be suitable. To solve this problem we transform the matrix of citation so that the distance between two documents i and j is minimal. We use for that the algorithm Dijkstra [1] in order to determine the minimal distance between two document i and j.

2.2 Importation of annotations

The goal in this part is to import and present in a relevant way the annotations of the documents cited by a document d. The presentation of the imported annotations is made by defining a multi-criteria choice to select annotations to be used in the following phase.

We found the most important annotations in this order:

1. the annotations which come from the documents which are located in important clusters;
2. within the annotations which come from the same cluster or from clusters which have the same importance, the important annotations are those which come from the documents which have an important degree of membership of the cluster. In other words, the document annotation importance depends of the importance of the document in the cluster;
3. if, the annotations come from the same document, or of documents which then have the same degree of membership of same cluster we consider as important the redundant annotations.

3. EXPERIMENTS AND CONCLUSION

To experiment our approach we have considered the CiteSeer[1] collection as a test database. CiteSeer is a digital library for scientific literature. CiteSeer localises scientific publications on the Web and extracts some information such as citations, title, authors and so on. This collection has been selected for two reasons: *(i)* the important number of documents; *(ii)* the fact that it contains scientific documents that using several citations. We have built a database that contains more than 550 000 documents.

However, CiteSeer description of documents cannot be used directly. In fact, CiteSeer uses a general vocabulary to describe the content of a document. But, we were interested only in the description of documents using a controlled vocabulary or an ontology. We have used the DMOZ[2] controlled vocabulary as an ontology to annotate CiteSeer documents during the experiment.

The first annotations results appeared satisfactory. For the moment, our method evaluation is based on the judgement of experts of the field, by comparing the annotation of our system with that of the expert. The experiments with the CiteSeer database have shown the feasibility of the approach and have allowed the automatic annotation of scientific articles. However, we still need further evaluation approach independently from human experts.

4. REFERENCES

[1] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[2] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1974.

[1] http://citeseer.ist.psu.edu/
[2] http://www.dmoz.org/

Semantic Similarity of Ontology Instances tailored on the Application Context

R. Albertoni
CNR-IMATI-GE
Via de Marini 6, 16149 Genova (IT)
+39 010 6475697
albertoni@ge.imati.cnr.it

M. De Martino
CNR-IMATI-GE
Via de Marini 6, 16149 Genova (IT)
+39 010 6475692
demartino@ge.imati.cnr.it

The poster presents a framework to assess a semantic asymmetric similarity among the instances of an ontology. It aims to define measurement of semantic similarity, which takes into account different hints hidden in the ontology definition and explicitly considers the application context. The similarity measurement is computed by combining and extending existing similarity measures [1,2] and tailoring them according to the criteria induced by the context.

In this decade, the ontologies have been imposing in the computer science as artefact to represent explicitly shared conceptualisation. Methods to assess similarity among instances are needed to exploit the knowledge modelled in the ontology in different research fields pertaining the Knowledge Management such as Data Mining and Information Visualization. They should consider as much as possible the implicit information encoded in the ontology as they provide useful hints to define the similarity. Moreover, they should be sensible to specific contexts inasmuch as different contexts induce different criteria of similarity.

So far, the most of research activity pertaining to similarity and ontologies has been carried out within the field of ontology alignment or to assess the similarity among concepts. Unfortunately, all these methods result inappropriate for the similarity among instances. On the one hand the similarities for the ontology alignment strongly focus on the comparison of the structural parts of distinct ontologies, therefore their application to assess the similarity among instances might result misleading. On the other hand, the concepts' similarities mainly deal with lexicographic ontologies ignoring the comparison of the instances values. Apart from them, few methods to assess similarities among instances have been proposed. Unfortunately these methods rarely take into account the different hints hidden in the ontology and they do not consider that the ontology entities differently concur in the similarity assessment according to the application context.

To overcome these limitations the research described in the poster aims to demonstrate a new sensitive measurement of semantic similarity among instances. It is defined by an amalgamation function, which aggregates different similarity measurements considering hints lying at different levels such as the structural comparison between two instances in terms of the classes that the instances belong to, and the instances comparison in term of their attributes and relations. In particular it is characterised by two similarity functions named *external similarity* and *extensional similarity*.

The *external similarity* performs a structural comparison between two instances i_1,i_2 in terms of the classes c_1, c_2 the instances belong to. It consists of two similarity evaluations:

- Class Matching, which is based on the distance between the classes c_1, c_2 and their depth respect to the class hierarchy in the ontology.
- Slot Matching, which is based on the number of attributes and relations shared by the classes c_1, c_2 with respect to the overall number of their attributes and relations. Then two classes having a plenty of attributes/relations, some of whose are in common, are less similar than two classes having less attributes but the same number of common attributes/relations.

The *extensional similarity* performs the instances comparison in term of their attributes and relations. Its evaluation is parametric with respect to the assessment criteria induced by the context. In application context the criteria induced by the context are explicitly formalized considering the importance of the entities (attributes and relations), which concur in the similarity assessment and the operation to compare them. Through this formalization is possible to tailor the similarity to specific application need. All the details pertaining to the method are available in [3].

The proposed method has been applied to compare the members of research staff. A simplified version of the ontology KA[1] that formalises concepts from academic research (Fig 1) is considered. Two applications are considered: "comparison of the members of the research staff according to their working experience" and "comparison of the members of the research staff with respect to their research interest". These applications induce two distinct application contexts:

- "Exp" induced by the comparison of the members of the research staff according to their working experience. The similarity among the members of the research staff (instances of the class *ResearchStaff*[2]) is roughly assessed considering the member's age (the attribute *age* inherited by the class *Person*), the number of projects and publications a researcher has worked on (the number of instances reachable through the relation *publication* and relation *workAtProject* inherited by *Staff*).

[1] http://protege.stanford.edu/plugins/owl/owl-library/ka.owl

- "Int" induced by the comparison of the members of the research staff with respect to their research interest. The researchers can be compared with respect to their interest (instances reachable through the relation *interest*), and again the publications (instances reachable through the relation *publications*), the projects (instances reachable through the relation *workAtProject*).

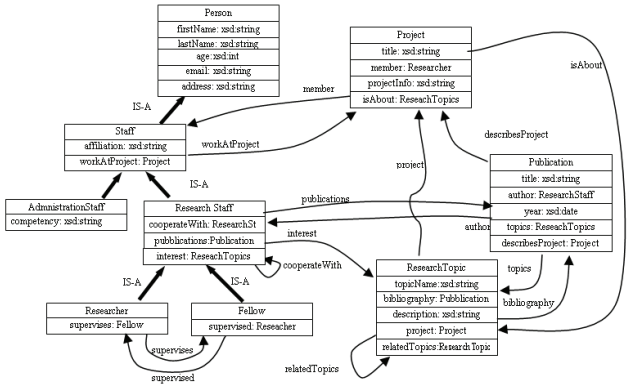
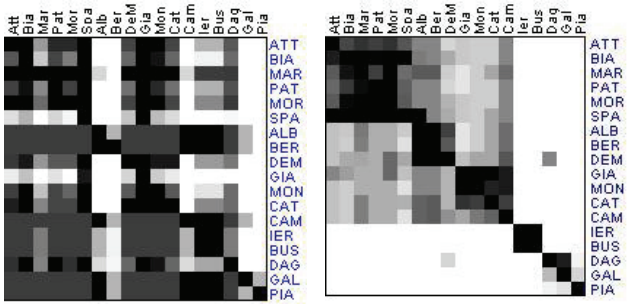


Fig. 1. Ontology related to the academic research.

The similarity assessment among the research staff working at the CNR-IMATI-GE is considered as application case. Two experiments are performed considering the two contexts "Exp", "Int". The information related the curricula of eighteen members of the research staff published at the IMATI web site (http//www.ge.imati.cnr.it) are used to populate the ontology.



(a) (b)

Fig. 2. (a) Similarity matrix for context "Exp"; (b) Similarity matrix for context "Int".

Fig 2 illustrates the results of the two experiments: (a) is the result related the context "Exp" and (b) is the result related to the context "Int". Each column j and each row i of the matrix represent a member of the research staff (identified by the first three letter of his family name). The grey level of the pixel (i,j) represents the similarity value (Sim(i,j)) between the two members located at the row i and columns j: the darker is the colour the more similar are the two researchers. Analysing the similarity matrices it is easy to realize that they are asymmetric; this confirms that the proposed model assesses an asymmetric similarity. Comparing the two matrices, it stands out how they are different: it is evident that the two contexts induce completely different similarity values.

Two kind of evaluations of the result concerning the similarity obtained with respect to the research interest (**Fig. 2**.b) are performed. The first evaluation is based on the concept of recall and precision calculated considering the same adaptation of recall and precision made by [2]. More precisely, considering an entity x the recall and precision are defined respectively as $(A \cap B)/A$, $(A \cap B)/B$ where A is the set of entities expected to be similar to x, and B is the set of similar entity calculated by a model. A critical issue in the similarity evaluation is to have a ground truth with respect to comparing the results obtained. We face this problem referring to the research staff of our institute and considering "similar" two members of the same research group. The average recall is estimated equal to 100% with a precision of 95%. These results are quite encouraging: the recall equal to 100% demonstrates that for each research group the similarity is able to rank all the expected members while the precision equal to 95% means that the average number of outsiders to be considered to rank all group members is equal to 5%.

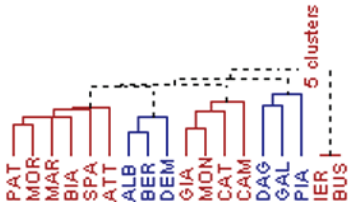


Fig. 3. The dendrogram obtained by the hierarchical clustering.

A second evaluation of the result related the application context "Int" is performed characterised by a data mining application. For each researcher and fellow we have computed his similarity with respect to the other members applying our method. In this way, we associate to each research staff member a string of values, which correspond to his relative distance from the other members. The strings correspond to the rows of the similarity matrix (**Fig. 2**.b). Then we have applied a tool to perform the hierarchical clustering among genetic micro array to the set of strings, considering each string as a kind of researcher genetic code. The dendrogram obtained is shown in **Fig. 3**, it recognizes the five clusters, which resemble the research group structure of our institute.

ACKNOWLEDGMENTS

This research started within the EU founded INVISIP project and then has been partially performed within the NoE AIM@SHAPE.

REFERENCES

[1] Maedche, A. and Zacharias, V. Clustering Ontology Based Metadata in the Semantic Web. 6th European Conference in Principles of Data Mining and Knowledge Discovery, PKDD 2002.

[2] Rodriguez, M. A. and Egenhofer, M. J.: Determining semantic similarity among entity classes from different ontologies. IEEE Trans.Knowl.Data Eng. Vol. 15[2]. (2003) 442-456.

[3] Albertoni R., De Martino, M, Semantic Similarity of Ontology Instances Tailored on the application Context, to appear in ODBASE 2006.

Semi-Distributed Development of Agent-Based Consultation Systems

Georg Buscher Joachim Baumeister Frank Puppe Dietmar Seipel
Institute of Computer Science, University of Würzburg, Germany
{buscher, baumeister, puppe, seipel}@informatik.uni-wuerzburg.de

ABSTRACT
This paper proposes a semi-distributed approach for the development of agent-based consultation systems. The key idea is: what can easily be done to enhance quality, and reduce redundancy is done centrally, while the mass of knowledge is acquired in a distributed way.

Categories and Subject Descriptors
I.2.1 Applications and Expert Systems, I.2.11 Distributed Artificial Intelligence, H.3.3 Information Search and Retrieval

Keywords
knowledge acquisition, collaborative development, agent-based consultation systems, large-scale knowledge-based systems

1. INTRODUCTION
An agent-based consultation system consists of a society of agents offering consulting services to the user. It emphasizes the active role of the user during the problem solving process, since the user can select which agents to consult. While each agent delivers a useful consultation service on its own, they are designed to complement each other, so that they together may offer consultation services comparable to traditional monolithic consultation systems (MCS; see e.g. [1]). An example for a user-centered consultation system (UCCS) for diagnosis is presented in [2]. It allows for distributed knowledge acquisition from volunteer contributors [3].

Both UCCS and MCS have advantages and problems concerning knowledge acquisition. MCS suffer from the bottleneck of centralized knowledge acquisition making it nearly impossible to develop large systems. However, in particular in medical domains, where symptoms are usually highly ambiguous, large systems can provide a much higher benefit to the user than specialized systems. Large MCS suffer from the rigidity of the chosen knowledge representation, which usually does not fit all parts of the system equally well. A uniform knowledge representation also causes inflexibility concerning user interaction; adapting the knowledge for different user types further increases the already nearly prohibitive knowledge acquisition effort. UCCS on the other hand allow for a distributed development, since each agent has well defined interfaces to its environment (i.e. the other agents). There is a structured interaction in UCCS among the agents: For example, a symptom class agent suspects diagnoses based on entered symptoms, clarification agents validate a diagnosis suspected by symptom class agents or by the user and a therapeutic agent determines adequate therapies for a validated diagnosis. The agents may have knowledge representation suitable for their purposes. Another advantage of UCCS compared to MCS is that the system is already usable with only some agents available. However, the knowledge of the different agent types overlaps, because a single symptom-diagnosis relation is usually needed by a symptom class agent as well as by a clarification agent resulting in redundancy problems. In addition, the distributed development makes it difficult to ensure quality standards over all agents. Further on, some services of a MCS are difficult to achieve by a UCCS like systematic dialogue guidance enabling high quality documentation and subsequent data mining options.

As a result a combination of centralized and distributed development may help to overcome some of the weaknesses of the respective approaches. The key idea is: what can easily be done to enhance quality, and reduce redundancy is done centrally, while the mass of knowledge is acquired in a distributed way. The "knowledge champion" initializes the consultation system by providing a basic list of necessary agents and may suggest or define the coarse structure of the agents. He or she is also responsible for organizing quality control measures (e.g. based on peer review or feedback from users consulting the UCCS), initiating activities to overcome detected weaknesses and decide in controversies resulting from the distributed process.

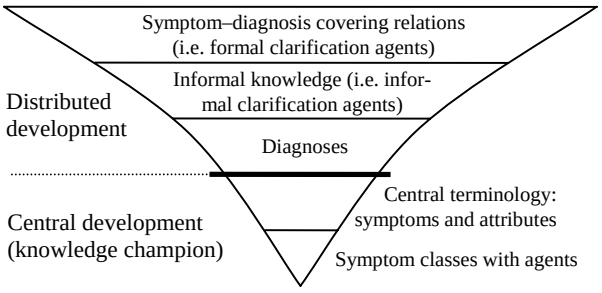


Figure 1. Development tasks and their effort.

Most of the concepts described in the following are exemplified by a pub recommendation system (PRS). It contains both a hierarchy of selection criteria (symptom classes) and of pubs (diagnoses). To every selection criterion, a formal symptom class agent is assigned, to every pub, a formal and an informal clarification

Conference'04, Month 1–2, 2004, City, State, Country.

agent exist. The development tasks accomplished in centralized and distributed development are presented in Figure 1.

2. ASPECTS OF SEMI-DISTRIBUTED DEVELOPMENT

There are several important issues concerning semi-distributed development of a UCCS being described in the following:

The *knowledge representation of the UCCS agents* has to be suitable for distributed development. In general, set-covering models and Bayesian Nets seem to be appropriate here due to the independency of the knowledge concerning different diagnoses. Heuristic rules on the other hand are more problematic for distributed development because the collaboration of the different rules concerning several diagnoses is a very sensitive process. For the formal agents of the PRS, for example, we chose set-covering knowledge, because the specification of set-covering relations between pubs and pub characteristics is a relatively simple process.

Avoiding redundancy of symptom class and diagnosis clarification agents: As stated in the introduction, the knowledge of the different agent types overlaps, because the same symptom-diagnosis relation is usually needed by a symptom class agent as well as by a clarification agent. The general idea to avoid redundancy is to generate the knowledge of symptom class agents from the knowledge of diagnosis clarification agents. This requires using the same terminology for both agent types. We achieve this in the PRS by predefining a centralized terminology (see Fig. 1; a task of the knowledge champion). Thus different developers add knowledge only for diagnosis clarification agents in a distributed but coordinated way.

Omitting *terminology alignments between different agents* is a further advantage of using a centralized terminology for all agents. Otherwise, due to the requirement that the agents should able to exchange meaningful data between each other, alignments between the different terminologies of different agents would be necessary and this normally includes problems concerning a loss of precision.

Modifications of the centralized terminology might affect all agents in the system. Removing a symptom involves removing the knowledge of all agents concerning this symptom. Adding a new symptom requires the insertion of default knowledge for this symptom to all agents. With respect to the PRS, this implies inserting zero set-covering relations concerning the different pubs and the new symptom, which means that the new symptom neither counts for nor against a pub. In addition, an alerting system is provided, which informs the agents' developers about such modifications.

Knowledge acquisition with the help of flexible templates can be applied to reduce inconsistencies of the distributedly entered knowledge and to simplify the knowledge acquisition process [4]. Concerning the PRS, for example, the developers can put in the characteristics of a new pub with the help of web-templates automatically generated from the centralized terminology. Thus, more people can be motivated to contribute to the system. Furthermore, wrong or inconsistent inputs can be avoided, which benefits the quality of the whole system.

Multiple opinions of the agents' developers are a typical problem in distibutedly developed systems. There are mainly three alternatives, how to handle this problem: First, the strict way would be to allow only one agent for every entry (i.e. symptom class or diagnosis), which cannot be modified by other developers than its author. Developers of other opinions only have the possibility to add comments for the agent. Second, in addition to the first alternative, every developer could be allowed to modify an agent. This might lead to faster development but might also yield unproductive controversies as they can sometimes be observed in www.wikipedia.org. Third, in addition to the first and second alternative, the developers could be allowed to add a new agent for the same entry in parallel stating a different opinion, which means that inconsistencies between agents for the same entry are explicitly tolerated. Currently, we experiment with all three alternatives in the PRS.

Feedback management is important to improve the quality of the different agents and therefore the whole system. The users have the possibility to give explicit feedback by rating an agent with a grade or writing a comment. This feedback is visible to all users and in particular to the authors of the respective agents, who might change the agents accordingly. In this way, an iterative development cycle emerges.

3. DISCUSSION

The proposed work can be viewed as a special realization of the semantic web idea. Instead of developing HTML- or Wiki-pages for pub recommendation directly, such pages are generated with a specific underlying semantic via knowledge templates and predefined terminologies in a distributed process with volunteer contribution. This knowledge is used by an inference engine and for generating a user-friendly search interface to increase precision and recall in comparison to standard search engines. We plan, if parts of the knowledge are already available in the Web (e.g. prices for beverages in the PRS), the knowledge base should be updated automatically by information extraction agents. Since the areas of application for UCCS reach from medical and legal advice via product selection to most diagnostic and classification domains, we also started other applications (e.g. MediSuggest for medical advise; www.medicoconsult.de; in German).

4. REFERENCES

[1] Huettig, M., Buscher, G., Menzel, M., Scheppach, W., Puppe, F., Buscher, H.-P.: A Diagnostic Expert System for Structured Reports, Quality Assessment, and Training of Residents in Sonography: Medizinische Klinik 99, 117-122, 2004

[2] Buscher, G., Baumeister, J., Puppe, F., Seipel, D.: User-Centered Consultation by a Society of Agents. Proc. 3rd International Conference on Knowledge Capture (K-CAP 2005), Banff, Canada, 27-34, 2005

[3] Symposium on Knowledge Collection from Volunteer Contributors (KCVC-05), AAAI Spring Symposium, technical report SS-05-03, http://www.aaai.org/Library/Symposia/Spring/ss05-03.php, 2005

[4] Chklovski, T., Gil, Y.: Improving the Design of Intelligent Acquisition Interfaces for Collecting World Knowledge from Web Contributors. Proc. K-CAP 2005, Banff, 35-42, 2005

QBLS: Semantic Web Technology for E-learning in Practice

Sylvain Dehors
Acacia, INRIA
2004 route des Lucioles
06902 Sophia Antipolis, France
sylvain.dehors@sophia.inria.fr

Catherine Faron Zucker
Mainline, I3S,
930 route des Colles
06930 Sophia Antipolis, France
faron@essi.fr

Rose Dieng-Kuntz
Acacia, INRIA
2004 route des Lucioles
06902 Sophia Antipolis, France
rose.dieng@sophia.inria.fr

ABSTRACT

With the spread of technology in education, more and more digital learning resources are being produced. The possibility to exchange them across the Web has brought the vision that accessing and reusing existing material would help both teachers and students. In parallel, *closed*, intelligent and adaptive systems have been developed, enriching interactions with learners based on explicit knowledge models. The semantic web proposal offers an opportunity to unify both ideas, but very few practical implementations have raised the challenge. In this paper we investigate the realisation of this proposal and give detailed information on the use of semantic web technologies in this context. Conclusions are based on real world experiences and the proposal is illustrated through the implementation of the QBLS system, currently in use at the local engineering school.

1. INTRODUCTION

The Semantic Web aims at exchanging information that can be understood by both humans and machines. The classical problems of e-learning: reusability, use of formalised knowledge, automatic adaptation, etc. should be answered by the Semantic Web. However the practical side remains largely unexplored. In this paper we show how the semantic web approach (using its languages and tools) benefits to the design and implementation of intelligent learning systems. Benefits are presented for the architecture, for the expression of models (domain, pedagogy) involved in retrieval and reasoning tasks and for user adaptation based on ontological commitments.

Results are backed by the implementation of a learning system completely relying on semantic web standard formalisms and technologies: The QBLS (Question Based Learning System) reusing a large coherent set of resources taken from the web to help students perform assignments. It has been effectively used in different real-world experiments at the EPU school of Nice.

2. Architecture for a SW-Learning System

To implement an architecture centred on the integration of a semantic search engine, as opposed to a multi-canal approach, the idea is to rely on a single entry point for semantic information and create what is called a "semantic middleware". The Corese search engine [1] is used in this scope. It takes ontologies and annotations (in OWL and RDF) and answers semantic queries (in SPARQL[4], the future W3C recommendation).

The QBLS system is deployed on a web server and accesses an instance of the Corese semantic search engine. HTTP requests are answered through JSP pages and servlets accessing the engine to build dynamic answers. The learning resources are XHTML pages, stored on the server. This standard format allows the system to manipulate and perform adaptation on the content. XSL transformations are used to construct the interface by combining the resources and the results from the engine. All the inferences are dynamically performed by the semantic search engine. Currently, QBLS has been deployed in three different experiments: (1) a 2 hour assignment session on signal analysis [2], (2) as a service to the ASPL platform of the Knowledge Web NoE, and (3) during the first semester of introductory teaching of Java programming.

3. Ontology based selection and sequencing

Resources consist of slides available on the web. RDF annotations link resources to domain and pedagogical concepts. Concepts are defined in various ontologies: domain, pedagogy and document model. Annotations are manually added on the documents, using styling features in OpenOffice, and automatically translated to RDF. The system keeps track of the user activity by generating other RDF triples, dynamically added to the annotation base.

Domain annotation classically consists of linking resources to domain concepts through "subject" relationships. Accordingly, typical queries follow the pattern: "give me resources describing the concept X". Thus we distinguish two ways of using domain knowledge in a "semantic" learning system: with a pre-defined domain ontology (for ex. in OWL), or with a less constraint domain vocabulary like in adaptive hypermedia. When looking at how these representations are used in practice, both identified approaches can be supported by semantic web technologies. In the first case, the ontology is mostly used to guide user through its subsumption hierarchy (e.g. finding the documents that are subjects of specialisations of a given domain concept). In practice, few other inferences than transitivity along subsumption links are used. When working with less constraints models, the SKOS [3] meta-model offers specialisation/generalisation relations between concepts (narrower/broader). SKOS and RDF then perfectly handle most graph models of adaptive hypermedia in a much natural way. For example, in Java the keyword "if" can be defined as a narrower concept of "conditional statement" whereas it is not a sub-class of it and such link must be modelled specifically in an ontology. This complicated the inferences

Poster and Demo of the 15th International Conference on Knowledge Engineering and Knowledge Management EKAW 2006 Podebrady, Czech Republic, 2nd-6th October, 2006.

whereas only transitivity along broader links needs to be computed by the chosen search engine.

To measure the impact of the model on the retrieval mechanism, using only direct linking between the 164 concepts and 443 resources (experiment 3), we found that 2.4 resources (avg.) per concept is returned. Exploiting transitivity along the links gives 2.9. This indicates a significant improvement of recall (the manual annotation already guarantees a 100% precision). As interfaces are usually web based, navigation follows a hypertext mode, so retrieving more resources also means getting more links to navigate from. We notice an increase from 3.1 to 3.7 in the number of offered directions from a resource when using the inference along broader links. Finally graph-connectivity is increased, creating more conceptually coherent learning paths.

Another model used in learning system is the pedagogical knowledge, usually expressed in a distinct ontology. It is used by the system to guide the learner from a pedagogical point of view, and many uses exist. In QBLS, several resources are relative to each domain concept. An "intelligent" behaviour consist in exploiting external knowledge to plan a coherent path among these results. This is done by ordering the resources according to their pedagogical type defined in the ontology. We have observed that roughly 50% of the time, students visit the resources in the natural order of display from left to right. Thus half of the learning path can be pre-determined. The other half is the result of learner's choices, based on interface information (i.e. resource type and title).

Using the inference capabilities of the Corese search engine, resources can be sorted automatically according to this type because they are hierarchically organised in the pedagogical ontology and statements are added by an expert teacher, using RDF triples. For example, the following pedagogical expertise: "*fundamental resources are prior to auxiliary resources*" is expressed by: "edu:Fundamental edu:priorTo edu:Auxiliary". From a small set of those statements, and using forward chaining rules, the semantic search engine complements the RDF graph to create priority relations between every couple of pedagogic roles. Corese then can sort the results of the query according to an index based on these relations. The final query used is expressed in SPARQL [4] like this:

```
SELECT ?doc WHERE
?doc dc:subject ?y .
{?y =  java:Object UNION ?y skos:broader java:Object}.
?doc rdf:type ?y .
?y edu :order ?order
ORDER BY ?order
```

Through this example of applying rules to propagate pedagogical relations and using them to order concepts of the ontology, we show how generic inference mechanisms can rely on pedagogical information to suggest coherent paths to learners.

The resources and pedagogical ontology were reused from the web. The proposed mechanism should be seen has a generic solution for reusing a course, and developing any learning system involving semantic representations. In addition to the arguments taken from a general literature review, it already proved effective all along the course of our experiments.

4. User adaptation

Adaptation to learners represents an important added value of e-learning. Guiding the learner should reduce the cognitive efforts induced by hypertext navigation. Generic frameworks perform adaptation like resource recommendation or link hiding based on history and user model. But they cannot be easily reused, due to proprietary models and formalisms. Through semantic web generic tools and languages, the same adaptive features are obtained with far more flexibility. For example, sorting the resources according to their type, as presented above, can be personalised for each user or profile. Precisely a specialisation of the priority relation is associated to each user (or profile). The sorting mechanism based on the generic priority relation also works with its specialisations. Navigation features, like history and back button, can be also semantically adapted to help the user browse the semantic space. Adaptive "link hiding" in the content, depending on the current context given by the chapter, is achieved by querying the engine for the relevant concepts given the resource and the chapter. The list of concepts is passed on to the XSL style-sheet generating the interface. Links in the resource content are then displayed only if they point to those concepts. In a nutshell, both deep adaptation mechanisms, like path recommendation, and shallow interface personalisation, like history, unvisited/visited items, etc., can be handled by semantic queries involving ontological knowledge. Efficiency and scalability are demonstrated by the QBLS implementations.

5. Conclusion

The QBLS experience demonstrates the possibility to reuse an existing course material taken from the web and operate it in an intelligent and adaptive system, based on semantic web standards and technology. The results presented here emphasise the reuse of existing tools (Corese, OpenOffice) and W3C standards. The implemented resource navigation system relies on an ontology-guided information retrieval mechanism. This generic mechanism for semantic web is transposed in the e-learning field to perform resource selection and organisation. The establishment of large formal ontologies for the domain is not compulsory as we have demonstrated that interesting behaviour can be obtained using SKOS. The given details on the implementation of pedagogical reasoning and user adaptation show the possibilities offered by those technologies. Finally, we started looking at semantic integration of other learning systems, and interoperability between ontologies (not presented here).

6. REFERENCES

[1] Corby, O., Dieng-Kuntz, R., Faron-Zucker, C. Querying the Semantic Web with the CORESE search engine. in *Proc. of the 16th European Conference on Artificial Intelligence* (ECAI'2004), pp 705-709, Valencia, 22-27, August, 2004

[2] Dehors, S., Faron-Zucker, C., Giboin, A., Stromboni, J.P. Semi-automated Semantic Annotation of Learning Resources by Identifying Layout Features. In *International Workshop SW-EL*, AIED'2005, Amsterdam, July, 2005.

[3] Simple Knowledge Organisation System (SKOS), http://www.w3.org/2004/02/skos/

[4] SPARQL Query Language for RDF, W3C, http://www.w3.org/TR/rdf-sparql-query

Growing World Wide Social Network by Bridging Social Portals Using FOAF*

[Extended Abstract]

Mária Bieliková
Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies, Slovak University of Technology
Ilkovičova 3, 842 16 Bratislava, Slovakia
bielik@fiit.stuba.sk

György Frivolt
Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies, Slovak University of Technology
Ilkovičova 3, 842 16 Bratislava, Slovakia
frivolt@fiit.stuba.sk

ABSTRACT

Portals providing the modeling of social relations among people became more and more popular. Although the existing FOAF (Friend of the Friend) ontology developed for modeling such social relations was introduced and enjoyed popularity, it is not used in such extent that it can be considered as World Wide Social Network. Such network can bring benefits (e.g., by providing useful analysis) only if it is sufficiently large. The problem is that no bridge exists between the social portals (often enjoying commercial popularity) and the FOAF backed by the Semantic Web projects. Our aim is to find ways to stimulate existing social networks grow to the World Wide Social Network. We propose a method for bridging the different social portals that is based on employing web page wrappers for generation the output in Semantic Web format (RDF), namely the FOAF and in such a way enable the World Wide Social Network. We believe that the FOAF might become a good basis for standard for giving the possibility to interconnect users on different portals representing the same real person.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services

Keywords

communities, social networks

1. INTRODUCTION

Social portals (such as `www.friendster.com`, `www.iwiw.net`, `www.hi5.com`) are web sites where users having accounts can classify other users of the system as friends, family members, schoolmates or other type of relatives [2]. That way the users form a social network of people often being situated on different locations. These portals enjoy popularity and provide the users a simple way how to look up friends or be contacted by old, often even forgotten relatives. People interlinked through social portals are often claimed to form so called virtual communities.

Social relations can be modeled by graphs. The basic use case for adding a new relation starts by signing a user b as known by the user a. The user b receives a notification about this act from the user a. The relation becomes confirmed after the user b's acceptance. This process yields new edge in the social network. However, by the increase of the popularity of social portals also the number of such portals increases. These portals use their own data representation invisible for other systems. Users of different portals cannot get connected as the current social sites do not offer such a possibility. Different users remain enclosed in different systems forming components of the social network.

Our aim is to propose a method for interconnecting so much islands as possible and thus to support evolving of social network similar to current information network presented on the Web. We call it *World Wide Social Network*.

2. SOCIAL NETWORKS AND FOAF

Description of persons' profiles using the Semantic Web format (RDF) provides a possibility to store the user and relation definitions distributed in machine readable way over the Web space. The major difference between description by RDF and description of virtual communities on social portals lays in the possibility to distribute the content and use it for various services aimed at analysis of created network independently. FOAF (`http://xmlns.com/foaf/0.1`, friend of a friend) is an ontology for describing information related to particular person in machine readable way. It is represented by RDF language. The FOAF description may contain "knows" relations, which enable to refer other person's FOAF description placed on the Web. FOAF ontologies are currently used mainly by the Semantic Web researchers. However, its simplicity and the feature of having the descriptions of different people distributed over the Web might encourage other users as well.

*This work was partially supported by the Slovak Research and Development Agency under the contract No. APVT-20-007104 and the the Scientific Grant Agency of Slovak Republic, grant No. VG1/3102/06.

Posters and Demos of the 15th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2006 Podebrady, Czech Republic, 2nd-6th October, 2006

2.1 Possibilities for social network growing

In order to enlarge and build the World Wide Social Network we exploit information on the personal relationships from the HTML based Web and introduce it to the Semantic Web. There are several possibilities for evolving the World Wide Social Network. First, adopted by the FOAF project is solely based on developing tools for simplification of the FOAF description creation. It assumes highly motivated people who are able to understand advantages of publishing their FOAF on the Web. This approach, however, is not sufficient for evolving social network that would incorporate enough users to be qualified as a world wide social network.

Second approach is to bridge existing independent social networks that resize within closed social portals with the social network evolved on the Web. This can be one way process, i.e. the social portal would publish alternative representation of its content in the form readable be the FOAF agent. We do not consider that there is currently enough motivation for social portal providers to devote effort to this particular issue (first the network should grow sufficiently to be able to show benefits of its analysis). Other way includes our proposal of developing wrappers able to gather information from social portals.

Another way to support of social network evolution is to incorporate the FOAF generators into information systems of organizations, which often generate templates for web pages of employees. An enrichment of these templates with the RDF description and publishing it on the organization's web server will immediately enlarge existing social network.

2.2 Wrappers for bringing social portals

As there are not enough highly motivated people who understand advantages of publishing their FOAF on the Web that would incorporate enough user profiles to be qualified as a World Wide Social Network, we propose to bridge existing independent social networks that resize within closed social portals with the social network evolving on the Web. This is a one way process, i.e. the social portal would publish alternative representation of its content in the form readable by the FOAF agent. We place the web page wrappers on a public server addressable by an URI. The URI is formulated by a specification of the social portal and the user for wrapping. The wrapper of social network site acts as a gateway for the World Wide Social Network content and social portals. The structure is depicted in Figure 1.

FOAF wrapper (center of the Figure 1) provides FOAF interface for the content stored on social portals. The little faces symbolize FOAF profiles. That way it acts as a gateway from the FOAF profiles (under the solid line) to the profiles stored on social portals (above the solid line). The wrapped FOAF profiles (faces directly connected to the FOAF gateway) cannot reference each other, but they can be referenced by other FOAF profiles existing on the web. Tools for analyzing and presentation of the FOAF world (right-bottom side of the figure) can benefit from enlargement of the social network by exploiting of wrapped personal profiles and relationships between people.

We developed an environment for creation of wrappers [3]. It consists of a designer and interpreter. The created wrappers are "hard-wired" for sites they were developed for and capable for wrapping of a specific site gives higher reliability compared to generic wrappers. The created web-page wrapper has a program which is interpreted by the wrapper interpreter. The language of the wrapper program has a tree structure. The vertices of the tree are instructions of the program. Every instruction can have several parameters depending on the type of the instruction. Instructions of the wrapper program are related to the web page loading, navigation in the web pages (tackling cookies, authentication, etc.), and extraction of data into variables or to output.

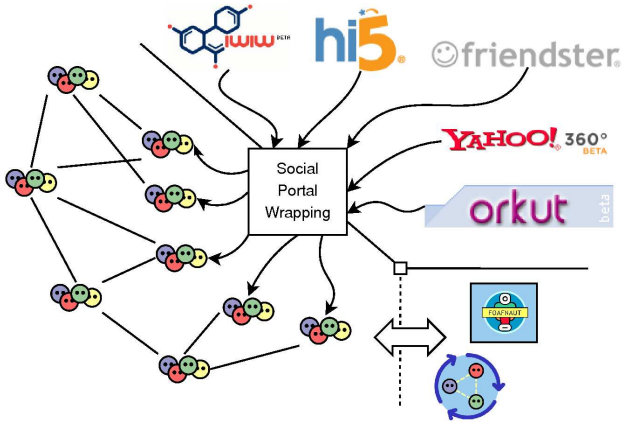


Figure 1: Using wrappers for enlargement of the World Wide Social Network.

3. SUMMARY AND FUTURE WORK

Motivation to publish personal profile on the Web is extremely important for evolving the World Wide Social Network. It is a chicken-egg problem. In order to have enough motivated people and services providing analysis of the network, large social network should exist. Our approach helps to overcome the issue of acquiring personal descriptions – to evolve existing isolated social networks into a World Wide Social Network. We developed an environment for designing wrappers. It helps in bringing user profiles from social portals to the FOAF world and in aggregating different profiles describing the same person under one FOAF instance. Evolution of the World Wide Social Network will bring several issues that open possibilities of further research, such as identification of different profiles that belong to the same person (e.g., transformed from several social portals), elaboration of a distributed model of social network wrapping or identification of vandalism on the World Wide Social Network, e.g. discovering profiles of non existing persons or profiles of real persons but with incorrect data.

4. REFERENCES

[1] K. Aberer et al. Emergent semantics principles and issues. In Y.-J. Lee et al editors, *DASFAA 2004*.

[2] Danah M. Boyd. Friendster and publicly articulated social networking. In *CHI '04*, pages 1279–1282, New York, NY, USA, 2004. ACM Press.

[3] P. Kasan et al. Automatized Information Retrieval from Heterogenous Web Sources. In M. Bieliková, editor, *IIT.SRC 2006*, pages 137–144. FIIT STU in Bratislava, April 2006.

A Semantic Web Approach to Software Maintenance

David Hyland-Wood
MIND Laboratory
University of Maryland College Park
College Park, MD, USA 20740
+1 301 314 6604
dwood@mindswap.org

David Carrington
School of Information Technology
and Electrical Engineering
The University of Queensland
Brisbane, Australia 4072
+61 7 3365 3310
davec@itee.uq.edu.au

Simon Kaplan
Faculty of Information Technology
Queensland University of Technology
Brisbane, Australia 4001
+61 7 3864 1913
s.kaplan@qut.edu.au

ABSTRACT
We propose an approach to software maintenance using Semantic Web techniques. Software system components and information about them (metadata) are represented in an ontology in the Web Ontology Language (OWL). Metadata includes functional and non-functional requirements documentation, metrics, the success or failure of tests and the means by which various components interact or were intended to interact. We discuss how this ontology, encoding of software system metadata in a Resource Description Framework (RDF) graph and SPARQL queries over the RDF graph can be used to enable language-neutral relational navigation of software systems to facilitate understanding and maintenance.

Categories and Subject Descriptors
D.2.1 [**Software**]: Software Engineering – *Requirements/Specifications.*

D.3.3 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods – *Semantic networks.*

General Terms
Algorithms, Management, Documentation.

Keywords
Software engineering, Software maintenance, Web Ontology Language, OWL, RDF, Semantic Web, Semantic network.

1. INTRODUCTION
Software systems and information about them diverge quickly in time, resulting in difficulties understanding and maintaining them. This divergence is typically a consequence of the loss of coupling between software components and system metadata [10]. In this paper we propose a methodology for capturing and making use of system metadata, coupling it with information regarding software components, relating it to an ontology of software engineering concepts (referred to hereafter as the SEC ontology) [3] and maintaining it over time. Unlike some previous attempts to address the loss of coupling [e.g., 2, 11], our methodology is based on standard data representations and may be applied to existing software systems. The methodology is robust in the sense that most of the required information may be automatically generated from existing sources, thus reducing the need for human input.

Semantic Web techniques include the Resource Description Framework (RDF) [7], Web Ontology Language (OWL) [6] and the SPARQL Query Language for RDF [9]. RDF is a language for representing information about resources, in particular information resources on the World Wide Web. We use it here to represent information about software components. OWL is a vocabulary description language for RDF. We use it here to define the relationships between software components, metrics, tests and requirements. These relationships are also defined in RDF, forming a graph of software components and their metadata with defined relationships between them. The SPARQL query language is then used to query this RDF graph in order to extract useful information regarding the state of the software system.

We chose to create an OWL ontology primarily to separate software engineering domain knowledge from operational knowledge (software components and system metadata), and to make our domain assumptions explicit. These have been identified as common reasons to use an ontological approach [8].

Our SEC ontology describes the relationship between object-oriented software components (programs which contain packages which contain classes, abstract classes and interfaces which contain methods and method signatures). The similarity to the language structure of Java is intentional, but eventual representation of C++, C# and other common object-oriented languages is desirable. Relationships captured include, for example, that an object-oriented class may implement an interface, extend a super class, contain methods, or have membership in a package.

Software tests, metrics and requirements are also represented in the ontology and their relationships defined to the various software components. Tests have results, denote the success or failure of the last run and the datetime of the last run. Tests are associated with software components and are themselves implemented as software components.

Metrics, like tests, are associated with a particular software component. They have values and datetimes when calculated. Descriptions (including units for the calculated metrics) are held in a generic RDF Schema comment annotation property.

EKAW 2006, October 2–6, 2006, Podebrady, Czech Republic.

Requirements are associated with multiple software components and can be encoded by one or more object-oriented classes. A particular method may be designated as the "entry point" for the requirement. An entry point provides a clue as to where to begin tracing the implementation in source code. The actual description of a requirement is provided in an RDF Schema comment.

A key to creating a graph useful for software engineering queries is capturing when information changes. This is done via an OWL object property `lastModifiedAt`, a datetime property that may be used on any software component, test, metric or requirement and denotes when it was last modified. SPARQL queries will rely on this information. Requirements have an additional datetime property to denote when they were last validated (by a human) against the software components that implement them.

Example data based on the SEC ontology was developed and is available online at [4]. The example data represents a small portion of a real-world software package. The example data consists of two object-oriented classes that contain four methods between them. They belong to a package, which belongs to a program. Each class has an associated unit test. A simple metric is associated with one of the classes. Each class has a requirement associated with it.

The example data was selected because it represented a small portion of a real code base. By developing SPARQL queries that returned useful information from the example data, the validity of the approach was shown.

The example data was loaded into the Redland RDF application framework [1] and SPARQL queries made against it. SPARQL queries were developed to show that properties representing the last modification of components and the last validation of requirements could be updated and that subsequent queries could be used to determine state changes. Queries were developed to show:

1. Whether or not requirements were currently validated against associated software components;
2. Which requirements required re-validation following a change to an associated software component;
3. Which tests have failed;
4. Which requirements relate to failed tests; and
5. Which object-oriented classes had associated tests.

These queries should be viewed as representative of the type of useful queries that can be made. The success of these SPARQL queries against real-world data show that Semantic Web techniques can be used to implement the relational navigation of software collaboration graphs and system metadata described in [5]. We believe that these techniques can be applied to existing systems (during reengineering, reverse engineering or routine maintenance). The mapping of requirements, metrics and tests to the elements of a software collaboration graph can occur at any time during a software system's life cycle.

The techniques considered in this paper may be implemented in an integrated development environment or project management tool. They require relatively little human input and would require little in the way of user interface intrusion.

Given that the life span of large software systems is limited by the ability of its maintainers to retain the links between system metadata and program elements, the potential benefit to further study of these techniques seems substantial.

2. ACKNOWLEDGMENTS

The authors wish to thank Paul Gearon of Herzum Software LLC, Brian Sletten of Bosatsu Consulting, Inc., Christian Halaschek-Wiener and Vladimir Kolovski of the MIND Laboratory for their kind suggestions for the improvement of the SEC ontology. David Hyland-Wood's efforts were partially funded by the National Science Foundation via the MIND Laboratory.

3. REFERENCES

[1] Beckett, D.: The Redland RDF Application Framework, http://librdf.org/ (updated 2006)

[2] Holt, P.O.: System Documentation and System Design: A Good Reason for Designing the Manual First, Proc. IEE Colloquium on Issues in Computer Support for Documentation and Manuals, (1993) 1/1-1/3

[3] Hyland-Wood, D.: An OWL-DL Ontology of Software Engineering Concepts, version 0.1, http://www.itee.uq.edu.au/~dwood/ontologies/sec.owl (2006)

[4] Hyland-Wood, D.: Example Data for an OWL-DL Ontology of Software Engineering Concepts, version 0.1, http://www.itee.uq.edu.au/~dwood/ontologies/sec-example.owl (2006)

[5] Jarrott, D., MacDonald, A.: Developing Relational Navigation to Effectively Understand Software., Proc. 10th Asia-Pacific Software Engineering Conference (APSEC'03) (2003) 144-153

[6] McGuinness, D., van Harmelen, F.: Web Ontology Language (OWL) Overview, W3C Recommendation, http://www.w3.org/TR/owl-features/ (2004)

[7] Manola, F., Miller, E. (eds.): RDF Primer, W3C Recommendation, http://www.w3.org/TR/rdf-primer/ (2004)

[8] Noy, N., McGuinness, D.: Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory, ftp://ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-01-05.pdf.gz (2001)

[9] Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF, W3C Candidate Recommendation, http://www.w3.org/TR/rdf-sparql-query/ (2006)

[10] VanDoren, E.: Maintenance of Operational Systems – An Overview, Carnegie Mellon Software Engineering Institute, http://www.sei.cmu.edu/str/descriptions/mos_body.html (1997)

[11] Van Lamsweerde, A., Delcourt, B., Delor, E., Schayes, M.-C., Champagne, R.: Generic Lifecycle Support in the ALMA environment, Proc. IEEE Transactions on Software Engineering, Vol 14, Issue 6, (1988) 720-7

Using a Semantic MediaWiki to Interact with a Knowledge-Based Infrastructure

Ian Millard, Afraz Jaffri, Hugh Glaser, Benedicto Rodriguez
School of Electronics and Computer Science
University of Southampton, SO17 1BJ, UK
{icm, aoj04r, hg, br05r}@ecs.soton.ac.uk

ABSTRACT

Facilitating knowledge acquisition is a task that usually requires special purpose interfaces with which users are not familiar. Providing effective acquisition through a familiar interface, such as a wiki, can provide a route to acquiring knowledge for low user investment. We present an architecture that is being used in the ReSIST project based on a Semantic MediaWiki integrated with a knowledge base that allows users to add and view knowledge using normal Semantic MediaWiki syntax. The architecture aims to facilitate the acquisition and representation of knowledge about resilient systems for users with no experience of knowledge technologies.

Categories and Subject Descriptors

D.4.3 [**Information Systems Applications**]: Communications Applications—*Information Browsers*

General Terms

Management, Design

1. INTRODUCTION

ReSIST [3] is a Network of Excellence which integrates leading researchers from the multidisciplinary domains of dependability, security, and human factors. The general focus of this activity is the advancement and development of technologies which will ensure that future ubiquitous computing systems have the necessary properties of resilience and survivability for real world deployment. The project also aims to create architectures which are tolerant of residual development and physical faults, interaction mistakes, and malicious attacks or service disruptions.

In an effort to aid the creation of such systems, the ReSIST project has embraced the emerging principles of Ontological Engineering and the Semantic Web. This has enabled us to formally describe resilience concepts and the properties of complex components in detail, as well as information regarding people, projects and publications. Through the utilisation of these semantic representations the ReSIST project is tasked with the creation of a Resilience Knowledge Base (RKB). The RKB will combine disparate information sources with suitable user interfaces to provide a central repository which comprehensively covers all aspects of resilient computing and dependable systems. It is envisioned that the RKB will provide an invaluable resource for both researchers and students.

The RKB is intended to provide information regarding organisations that are researching resilient systems; researchers interested in resilient systems; papers associated with resilient systems; faults, errors and failures that have occurred on IST systems; and other aspects of resilient systems research topics. In addition, knowledge regarding the ReSIST project itself is recorded, including sub-project activities, meetings, work package development and management decisions.

However, the task of acquiring semantic information about an ongoing project from people who are not experts in the field of knowledge related technologies presents a significant challenge. Systems must be provided to facilitate as much incidental knowledge acquisition as possible, while still being able to gather sufficient knowledge to be meaningful to the project.

2. SEMANTIC MEDIAWIKI

A significant step in achieving incidental knowledge acquisition has been through the use and customisation of the newly developed Semantic MediaWiki (SMW) [5]. In addition to supporting general collaboration between project members, the SMW provides a means of adding metadata about the concepts and relations that are contained within the wiki. This form of 'tagging' makes it relatively simple to turn such annotations into subject, predicate, object triples that can be stored as RDF and incorporated into the RKB. Such a system has the advantage of being easy to use for non experts, but also powerful in the way in which knowledge can be created and stored.

A prototype system has been developed, utilising the SMW in conjunction with an external 3store [4] RDF repository. In the SMW, real-world or abstract entities are represented by an individual page, to which metadata can be added. The page is therefore represented as the subject resource in RDF triple form. Relations and attributes are handled differently

Posters and Demos of the 15th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2006 Podebrady, Czech Republic, 2nd-6th October, 2006

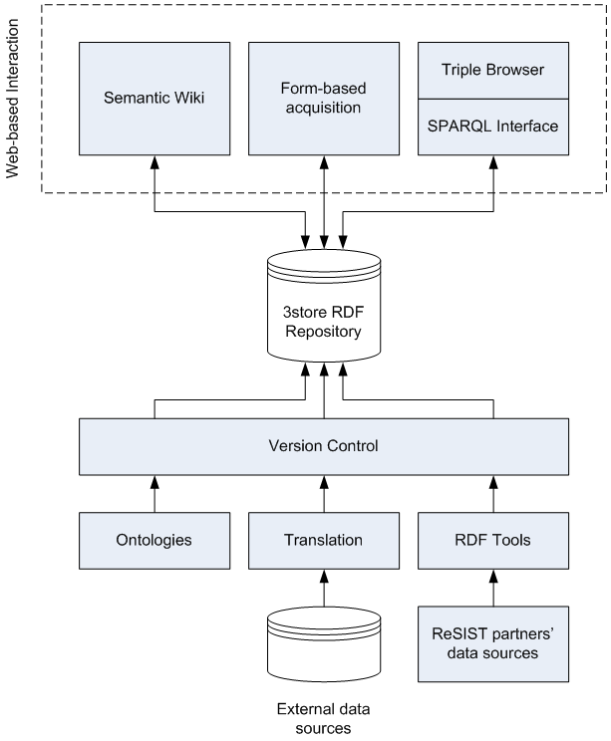


Figure 1: ReSIST RKB Architecture

within the SMW, associating the current page with another SMW resource or a literal value. When a page is saved in the SMW, custom code is invoked to export the relations and attributes as RDF and to assert them within the RKB repository.

For example, a page in the SMW may describe a publication, to which a project member wishes to be associated as an author. Utilising the AKT Ontology [1], the following relation may be inserted into the page, specifying the desired fact.

```
[[has author::User:Joe Bloggs|Joe Bloggs]]
```

Entering this special SMW markup within the page will cause the following triple to be asserted into the RKB:

```
<http://resist.eu/publications/Bloggs06>
<http://www.aktors.org/ontology/portal#has-author>
<http://resist.eu/people/Joe-Bloggs> .
```

However, one area in which the SMW lacks good support is that of namespaces, as general facilities for utilising external ontologies, concepts and data-types are yet to be implemented. In a closed world SMW deployment this is not a problem, and indeed simplifies the input required by users. Nevertheless, namespaces are vital for disambiguation and ontological inference, so the export routines apply the relevant namespace prefixes during RDF generation. This is achieved by using a static mapping between the SMW representation of ontological concepts and their external 'real-world' form in the RKB.

The SMW can also be used as a means of exposing knowledge stored within the RKB. For example, pages describing the classes and properties from external ontologies have been imported into the SMW, permitting users to view and discuss the rationale behind each. As well as facilitating collaborative ontology development, these representations allow users to readily see whether relations and concepts have been used appropriately when entering semantic markup.

In addition to knowledge obtained through the use of the SMW, significant efforts are underway to facilitate the acquisition of semantic metadata from external sources. Tools have been developed to allow non-expert users at each of the 18 ReSIST partner sites to periodically generate RDF data regarding their institution and its activities. This information is then 'pushed' to the RKB server through a version control mechanism and automatically asserted, maintaining an up-to-date representation of the disparate information sources. Work has also been done to allow the bulk-import of information from Cordis, the ACM publications database, Citeseer and smaller-scale EPrints repositories.

Finally, a generic web-based form interface has been developed which can be configured to allow the acquisition of information into a specific ontologically mediated format. This interface is currently being used to collate user-submitted data regarding university courses taught to students that are related to various aspects of resilient and dependable systems.

3. FUTURE WORK

The ReSIST project is currently in its ninth month and the benefits of using the RKB architecture can already be seen. Content acquisition will be an ongoing process, and should enable more interesting analysis to be performed once a more substantial data-set is available. Combined with this effort is a requirement to develop interfaces with which novice users can easily explore the RKB, which may potentially include extensions of the work demonstrated in the CS AKTive Space project [2]. However, the maintenance of large semantic data sets presents its own challenges, not least of which are issues regarding referential integrity of knowledge acquired from multiple sources.

4. ACKNOWLEDGMENTS

This work is supported under the ReSIST Network of Excellence, which is sponsored by the Information Society Technology (IST) priority in the EU Sixth Framework Programme (FP6) under contract number IST 4 026764 NOE.

5. REFERENCES

[1] AKT Ontology. http://www.aktors.org/ontology/.

[2] CS AKTive Space. http://cs.aktivespace.org/.

[3] The ReSIST Project. http://www.resist-noe.org/.

[4] S. Harris and N. Gibbins. 3Store: Efficient bulk RDF storage. In *Proceedings of the 1st International Workshop on Practical and Scalable Semantic Systems*, pages 1–15, 2003.

[5] Semantic MediaWiki. http://meta.wikimedia.org/wiki/Semantic_MediaWiki.

Ontological Concepts Evaluation Based on Context

Lobna Karoui
Ph.D Student
Ecole Supérieure d'Electricité
Plateau de Moulon 3 rue Joliot Curie
91192 Gif-sur-Yvette cedex FRANCE
Lobna.Karoui@supelec.fr

Marie-Aude Aufaure
Professor Assistant
Ecole Supérieur D'Electricité
Plateau de Moulon 3 rue Joliot Curie
91192 Gif-sur-Yvette cedex France
Marie-Aude.Aufaure@spelec.fr

ABSTRACT
Ontology evaluation is vital for the development and the deployment of many applications like data annotation, retrieval information and semantic Web. In this paper, we focus on the ontological concept evaluation task. We propose a new evaluation method based on a large collection of web documents, several context definitions deduced from it and an algorithm which computes the credibility degree associated to each word cluster and to each context. Our evaluation method permits to help the expert and gives the possible word associations existing in these contexts, some semantic tags suggestions, delete the noisy elements or move them to their appropriate cluster. Our experiments are conducted on French documents related to the tourism domain. The first experiments elaborated with experts' contact show how our method helps them and facilitates the evaluation task.

General Terms
Verifying.

Keywords
Context, ontology, evaluation, semantic web.

1. INTRODUCTION
Some current work in data annotation, data integration, information retrieval, building multi-agents application, semantic web services depends on ontologies. The development and the deployment of these applications are related to the richness of the conceptualization inside the ontology. Ontology [1] is "an explicit formalization of a shared understanding of a conceptualization. Many researchers are interested in the ontology evaluation [2, 3] i.e the evaluation of the concepts, relations among them, etc. For instance, in order to evaluate the vocabulary, Meadche and Staab [2] proposed an approach which aims to evaluate the lexical and vocabulary level of an ontology. They have defined a similarity measure in order to compare two strings one provided from the produced ontology and the other from an existing ontology. In [3], the authors evaluate their lexical by using WordNet and the notions of 'precision' and 'recall'. Based on this vocabulary, ontology building approaches applying clustering methods permit to obtain word clusters as potential future concepts. In this paper, we focus on the evaluation of the ontological concepts that are extracted from the web documents. We work on French documents related to the tourism domain. Our evaluation method is based on the concept of "Contextualization" and on a large collection of web documents. After treating and analyzing the documents according to our pre-processing step of our system [5], we obtain four files in which we find the nominal, verbal, prepositional and conjunctional groups. This information constitutes the linguistic context. Afterwards, our other process returns the four files where we find sections for each of the following sentences, paragraphs and documents. This second source of information represents the documentary context. Then based on these two context types, we define an algorithm which computes the credibility degree associated to each word cluster and to each context. Our algorithm titled "Credibility Degree Computation" and noted CDC permits either to help an ordinary user to evaluate the word clusters before the domain expert do it or the expert himself. It permits to give the possible word associations existing in these contexts, some semantic tags suggestions, delete the noisy elements or move them to their appropriate cluster. The CDC algorithm informs about the initial words of a given cluster and facilitates the evaluation task. Our evaluation method does not depend on a gold standard and it could be applied in any domain. Thanks to the quantitative and qualitative evaluation criteria, the first experiments elaborated with experts' contact show how our method helps them and facilitates the evaluation task. In the following section, we explain our concepts evaluation method.

2. CONTEXT-BASED CONCEPT EVALUATION
Our idea is as follows: "looking in the Web in order to understand the meaning of each word or two words together and so on" could be a solution but why? This task is a contextualization [4] operation. During the concept extraction task, terms are selected from their context in order to group them but they are presented to the knowledge engineer or the domain expert without any context after a decontextualization [4] process that's why the evaluation step is always difficult. In our case, for each cluster the general context is the domain (tourism). But this information is not sufficient to evaluate a cluster and to give it a semantic tag. A possible solution for ensuring easy analysis is using a big collection of web documents related to the same studied domain. The collection is written by persons having different opinions and purposes. In this case, domain vocabulary and situation deduced from it are varied. To obtain this domain web collection of French documents, we use a cleaner (HTTrack Website Copier). Then we treat them thanks to the pre-processing step of our system [5]. This step provides a set of programs using to clean and the information source (deleting some elements such as scripts, tags, and correct some codification). Also, our system offers analyses which are structural analysis, nature analysis and linguistic analysis. After these processes, we obtain a clean corpus useful for the rest of the application. Based on this web collection, we generate several contexts. A context is an appropriate support for a semantic interpretation i.e it limits the associated knowledge of each word and gives a background for the evaluation and labeling

task. In order to explain this idea, we take the sentence: “the possible accommodations in the east region of USA are hotels and residences. Within this example, when we limit the context to the association of ‘hotels’ and ‘residences’ by the conjunction ‘and’, we deduce that ‘hotels’ and ‘residences’ belongs to the same concept. However, when we limit the context to the entire sentence, we can say that the associated concept to these two words is ‘accommodation’. So, thanks to the contextualization task, we can deduce either the meaning of each word, or the semantic association between some words or the concept associated to some words. Taking into account a static context i.e only one such as a sentence for all the word clusters is not sufficient since in some case the sentence does not contain all the words of a cluster. That’s why, our evaluation is not restricted to a unique context on the contrary it depends on various granularity levels which are applied and considered consecutively. The several contexts defined from the domain web documents are provided by two sources. The first one is a linguistic analysis that permits to give us the various nominal groups and verbal groups. Also, it procures the various word associations by a preposition (of, on, etc.) or a co-ordinating conjunction (and, or, etc.). The second source is a documentary analysis that permits to give us the various sections of phrases (part of a phrase finished by a punctuation like ‘;’ or ‘,’), the sentences, the paragraphs and the documents. So, we have two types of contexts which are a linguistic context and a documentary context. By using the first one, we obtain the close words of our target terms. By using the second one, the context is more generalized than the linguistic one and the information deduced will be either complementary information or completely new information for the words of a cluster. Now, the problem is that the expert is not capable; even we give him all the analysis results, to find the possible association of the targets words especially that we work on a big corpus. In order to facilitate this process, we define a semantic index which represents the credibility of the target words’ association in relation with the different contexts. This index is named “credibility degree”. It computed for each word cluster and for each context definition in an automated way.

Our « Credibility Degree Computation » algorithm is executed on a set of word clusters in order to compute their credibility degree. Let us take an example in order to explain our idea: with the following word cluster {academy, golf, golfer, club} and according to one context definition (for example a sentence), the algorithm finds all the possible combination in the context i.e tries to find the four words (academy, golf, golfer, club), then the association of three words and so on. For each found association, it presents the associated words and gives a degree representing the number of times this type of association is found in the corpus and particularly in the same context’s type. For instance, with the same example, it finds two possible association with three words which are {academy, golf, golfer} and {golf, golfer, club} so the credibility degree is 3_2 i.e two associations of three words.

Our algorithm has several functionalities which are: finding the associations between some words in order to facilitate the labelling step, finding in the same time the available association in the context and the concept, detecting the noisy elements in a cluster and either delete them or move them to another cluster, enhancing a cluster by other words from the associations.
Our results obtained using our algorithm are presented to the user in two forms which are a HTML format and an evaluation railing.

Thanks to the credibility degrees computed for each cluster and for each context, the user obtain an amount of information useful and in some cases sufficient to manipulate (delete word, remove word, etc.), evaluate and label the cluster. For example, for a same cluster, if he find the three credibility degrees (5_1, 4_3, 3_8, 2_{15}), he begin analysing the association with 5 words. If it is not sufficient, he analyses the three associations of four words and so on. If the information returned by our algorithm to this cluster and for one context is not enough, he can look to the other credibility degree provided by the other contexts. Our concept evaluation method, based on a large collection of domain web documents and several contexts definitions with different granularity degree, permits to an ordinary user to help the expert by manipulating the word clusters and giving him semantic tags as a suggestion. Consequently, the expert should decide on the appropriateness of these labels as well as clusters homogeneities which are not labelled. It provides a quantitative evaluation, thanks to the credibility degrees for each cluster and for each context, and a qualitative evaluation thanks to the various word associations procured by our context refining process. The experiments of our evaluation method show how the contextualization process permits to help either the novice or the expert.

3. CONCLUSION

Ontology evaluation task is not evident. In this paper, we have proposed a new evaluation method that permits to help either an ordinary user (like a student or a knowledge engineer who are not specialized in each domain) or the domain expert to take the write decision about the semantic homogeneity of a cluster. In order to achieve this purpose, we have defined a new algorithm tilted Credibility Degree Computation noted CDC. Our algorithm tries to eliminate or remove the noisy elements, propose some semantic tags and give several word associations. Our method guides the expert to an easier interpretation of the word cluster and to avoiding the ambiguous cases. Future research in this area should seek to develop further techniques for evaluating the other elements of an ontology such as the relations between the concepts. Also, we define a contextual model according to the evaluation task and to develop an application that permits to combine or to make intersection between the results of two or many contexts analysis.

4. REFERENCES

[1] Gruber, T.R.,: *Towards Principles for the Design of Ontologies used for Knowledge sharing.* In N. Guarino and R. Poli, editors, Formal Ontology in Conceptual Analysis and Knowledge Representation, 1993

[2] Meadche, A and Staab, S., *Measuring similarity between ontologies*. Proc. CIKM 2002. LNAI vol.2473.

[3] Brewster, C., Alani, H., Dasmahapatra, S. and Wilks, Y., *Data Driven Ontology Evaluation*, Proceedings of Int. Conf. on language resources and evaluation, Lisbon, 2004.

[4] Brézillon, P.:"*Context in problem solving: A survey*", The Knowledge Engineering Review, Volume: 14, Issue: 1,Pages: 1-34, 1999.

[5] Karoui, L., Bennacer, N. and Aufaure, M-A.: *Extraction de concepts guidée par le contexte*. To appear in Proc. XIIIème Rencontres de la Société Francophone de Classification SFC'06, 2006.

Click stream analysis – the semantic approach

Tomas Kliegr
Department of Information and Knowledge Engineering
Faculty of Informatics and Statistics
University of Economics Prague
W. Churchill sq. 4, 130 67, Prague, Czech Republic
xklit05@vse.cz

ABSTRACT
This case-study driven research focuses on analyzing clickstreams - the trails visitors make when browsing a certain website. The approach taken is intended to overcome three main drawbacks of using association rules mining in clickstream analysis - high granularity of input (and consequently output) data, too weak patterns, and problematic transformation of variable-length clickstreams to a fixed number of attributes. Semantic information about visited pages is used to identify the main area of interest for each visitor. The generalized clickstreams are then in the form suitable for datamining, e.g. with the LISp-Miner system.

Keywords
Clickstream analysis, association rules, data-mining

1. INTRODUCTION
This research follows the CRISP-DM methodology and as such is case-study driven. The methods presented further originated as a response to a business need for a more comprehensive clickstream analysis solution for e-commerce.

The study of available literature (e.g. [1, 2]) indicated that more comprehensive analysis can be achieved primarily by involving more information into the datamining process (DM). Using sophisticated datamining techniques alone on sole clickstream data is not sufficient.

Approach presented in this paper is built primarily upon closing the semantic gap, but novel datamining techniques are also utilized. It draws into the DM process semantic information about the visited pages. The semantic meaning of each visit is approximated by 30 attributes, which form the *visitor profile*. These attributes were designed to convey the visitor's purpose during the visit. Each visit can consist of any number of visited pages. However, many systems for mining association rules can work only with data represented by a fixed number of attributes (i.e. columns in a table). Visitor profiles introduced in Chapter 3 effectively solve this problem. Data prepared in this way can be mined with any system for mining association rules.

Due to limited space, the semantic related aspects of the research are highlighted.

2. OBTAINING DATA
In order to carry out the clickstream analysis two distinct kinds of data are needed. The actual clickstream data and the semantic meta-data for the visited pages.

Collecting clickstreams Although tracking page views on the application layer is becoming increasingly common and recommended [2], it might be still useful to highlight the main advantages it has over (more traditional) log-file based approaches: a) Automatic robot filtering, b) Fewer anomalies, c) Storage efficiency.

Robot filtering is achieved by the fact that the tracking application is only activated by the visitor downloading a tracking picture. Anomalies are avoided, because sessions are formed automatically (in log-file based approaches heuristics have to be usually applied to form sessions). The fact that clickstreams are saved in a sessionized state straight into a relational database produces substantial savings in disk space compared to redundant information in log files.

Obtaining semantic data Figure 1 indicates that each page is assigned a content descriptor (e.g. Alps) and a service descriptor (e.g. Search page). The assigned descriptors should preferably form a taxonomy (e.g. catalog page as a special kind of search page). A similar approach is presented in [4].

The content taxonomy was created by a domain expert. It contains a list of categories (taxonomy of the travel agencys website), and a list of product IDs belonging to each category. Another domain-expert (web master) created a parallel taxonomy describing the service taxonomy.

These taxonomies became a basis for the visitors profile. Both the ontology and the mapping was stored in a proprietary formatted database.

3. PREPARING DATA
The collected click streams have varying lengths. The goal of this phase was to create a fixed-length *visitor profile* based

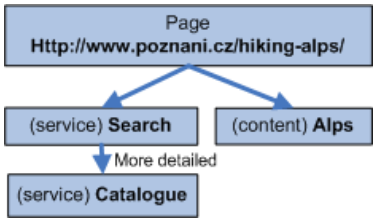


Figure 1: Sample page classification

on a generalization (conceptualization) of the click stream. The *visitor profile* should be constituted by an appropriately chosen set of attributes in a way that would represent the source data a with minimal loss of information.

Classifying pages First, pages were classified into the content and service taxonomies. Pages in clickstream were matched into service and content taxonomies and also with some other attributes: Time on page (t), Order of the page in the click stream (o) and Score (S).

Score was designed to express the absolute weight of a particular page in the visitors path. Its formula takes into account the time on page, but a higher weight is given to pages with higher order number. Following is an experimental ad hoc formula used to compute Score.

$$S = (\ln(o) + 1) * t \quad (1)$$

Visitor profile The last step of the data preparation phase was to create a fixed-length *visitor profile*. The attributes, which represent events that are common to all visits (e.g. Entrance Page or Number of Visited Pages), did not pose a problem and could be included to the profile straight away. However, the number of visited pages varies from visitor to visitor and must be thus pruned. The approach taken to pruning is based on creation of two derived attributes: the *Most favorite topic (MFT)* and the *Range of interest*.

The *MFT* estimates the purpose of an individual visit. It is important to note that the assumption is that the visitor had exactly one purpose. The *Range of interest* corresponds to the number of different topics viewed during a visit.

4. MODELING

A sample business analytic question can be "What are the visitors' interests in relation to the referring server?"

The data was analyzed using the *4ft-Miner* procedure of the *LISp-Miner* system [3]. The *LISp-Miner* procedures have the advantage that they offer various statistical tests (not only confidence and support). The *Above Average Implication Quantifier* $\Rightarrow_p^+$ (AAI) was used in the research. The advantage of this quantifier is that it captures patterns, which are not necessarily very strong, but are "above average". The nature of click stream data is such that non-trivial tight dependencies occurring in a substantial number of cases are rarely found. This is a reason why (AAI) produces superior results to some other information measures (e.g. confidence and support).

An association rule $Ant \Rightarrow_p^+ Suc$ can be interpreted such as *Among objects satisfying Antecedent there are at least 100*p per cent more objects satisfying Succedent then there are objects satisfying Succedent in the whole data matrix.*

Top 3 generated rules. Each rule is followed by a value of the Average difference - the AAI test criterion. The names of the websites were translated into English.:

1. $R(www.adventure.cz) \Rightarrow MFT(Expeditions)$, 22.218
2. $R(www.travelogues.cz) \Rightarrow MFT(Expeditions)$, 20.531
3. $R(www.hiking.cz) \Rightarrow MFT(ClimbingSchool)$, 17.733

These hypotheses indicate that some websites (e.g. hiking.cz) drive visitors, who are much more likely to be interested in a certain topic (Climbing school) than an average visitor.

5. FUTURE WORK

Future work will be aimed at involving full texts of the visited pages. Full texts can be for example used to aid generation of the service and topic taxonomies and new attributes expressing the relation between the query string, the content of the page and the behavior of a visitor during the visit.

It is suggested to use lemmatization and subsequently wordnet dictionary to find hypernyms for words contained in the query strings. For example it could be inferred that searches containing words Rome, Prague and London could be generalized to one term National capital.

6. CONCLUSION

The results obtained from the case study data showed feasibility of the introduced model, which is based on a conceptualization of visitor path and subsequent association rules mining using the *LISp-Miner* system. The article gave a brief outline of the proposals for further work which should be mainly aimed at generalizing the approach used in the case study as well as at involving full texts of the visited pages into the mining process.

7. ACKNOWLEDGMENTS

The work described here has been supported by the project 201/05/0325 of Czech Science Foundation, and by the project IGA 11/06 of University of Economics, Prague.

8. REFERENCES

[1] R. Cooley. The use of web structure and content to identify subjectively interesting web usage patterns. *ACM Trans. Inter. Tech.*, 3(2), 2003.

[2] F. M. Facca and P. L. Lanzi. Mining interesting knowledge from weblogs: a survey. In *Data & Knowledge Engineering*, volume 53. Elsevier, 2005.

[3] J. Rauch and M. Simunek. Alternative approach to mining association rules. In *The Foundation of Data Mining and Knowledge Discovery*, 2005.

[4] M. Vanzin and K. Becker. Exploiting knowledge representation for pattern. In *Proc. of the Workshop on Knowledge Disc. and Ontol.*, 2004.

RDQuery* - Querying Relational Databases on-the-fly with RDF-QL

Cristian Pérez de Laborda Matthäus Zloch Stefan Conrad

Institute of Computer Science
Heinrich-Heine-Universität Düsseldorf
Universitätsstr. 1
D-40225 Düsseldorf, Germany
{perezdel, conrad}@cs.uni-duesseldorf.de, matthaeus.zloch@uni-duesseldorf.de

ABSTRACT

One of the main drawbacks of the Semantic Web is the lack of semantically rich data, since most of the information is still stored in relational databases. We present RDQuery, a wrapper system which enables Semantic Web applications to access and query data actually stored in relational databases using their own built-in functionality. RDQuery automatically translates SPARQL and RDQL queries into SQL. The translation process is based on the Relational.OWL representation of relational databases and does not depend on the local schema or the underlying database management system.

1. INTRODUCTION

With his vision of a Semantic Web, Tim Berners-Lee inspired the database and knowledge representation communities to build up the next generation Web. Despite its sophisticated technologies like RDF [3] and OWL [4], the Semantic Web still has to face its major drawback, the lack of data. In fact, data is usually still stored in relational databases where it cannot be accessed directly by Semantic Web applications. Consequently, a well-defined mapping of relational to semantic data is required.

Although we can convert the schema of a relational database automatically into an RDF/OWL ontology and represent its data items as instances of this data source specific ontology [6], barely a database is static. Consequently, this data and schema extract may rapidly become outdated. Indeed, a schema or data extraction could be initiated, whenever a data or schema modification occurs within the database. Nevertheless, dealing with dynamic data sources, a direct access to such data sources would be preferable.

*RDQuery is published under GNU GPL and can be downloaded at `http://sourceforge.net/projects/rdquery/`.

Posters and Demos of the 15th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2006 Podebrady, Czech Republic, 2nd-6th October, 2006

2. RDQuery

RDQuery is a wrapper system which makes relational databases accessible for Semantic Web applications using an RDF query language (RDF-QL). RDQuery currently supports RDQL [9] and its successor SPARQL [8], which will hopefully be recommended soon by the W3C as the de facto standard query language for RDF. Nevertheless, RDQuery may easily be adapted to future developments adding specific parsers for other query languages. Figure 1 gives an overview of the RDQuery system architecture and depicts the path passed by a query until it reaches the relational database as its destination.

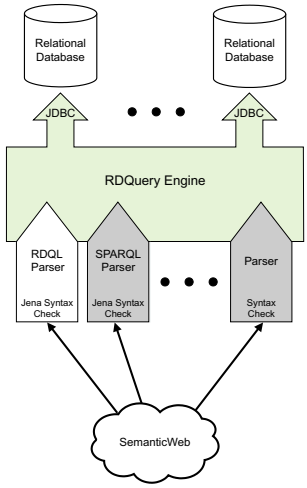


Figure 1: RDQuery System Architecture

First, the syntax of the query is validated and its relevant parts (e.g. the `WHERE` clause) are extracted using the built-in syntax checker of the JENA Framework [2]. Thereupon, the relevant parts of the query are once again parsed using an own JavaCC-based [1] grammar, in order to detect the properties of the query. Based on this information, the corresponding SQL query is built up. The resulting query is then executed and processed on the original database without having to translate the original database into a Relational.OWL representation, which thus only exists virtually.

The query translation is based on the results presented in [5] and [7], where we examined possible RDQL and SPARQL

correspondents for the basic expressions of the relational algebra. Each of the five main operations $\{\sigma, \pi, \cup, -, \times\}$ of the relational algebra has characteristic appearances within a Semantic Web query. A selection, i.e. the `WHERE` part of an SQL query, corresponds to a triple similar to `{?x dbinst:TABLE.COLUMN 'value'}`, where `?x` is a free variable and `TABLE.COLUMN`, the column where the `value` shall be matched. Similar mappings can be given for the remaining operations of the relational algebra.

Example: The SPARQL query

```
CONSTRUCT {?a ?b ?c}
WHERE {{?a ?b ?c}.
       {?a rdf:type db:customers}.
       {?a db:customers.City 'Berlin'}.
   FILTER (?b=db:customers.ContactName)}
```

is automatically recognized by RDQuery as the SPARQL correspondent of a selection, followed by a projection. It thus translates the given query automatically into the following SQL query:

```
SELECT customers.ContactName
FROM   customers
WHERE  customers.City = "Berlin"
```

After the query execution on the original database, the user may opt for an RDF processable representation of the query result. This feature of RDQuery is especially important for Semantic Web applications using a query language, which is not closed within RDF (e.g. RDQL), where the result of such a query is not a valid RDF graph, but a list of possible variable bindings.

The whole query transformation process is identical for any relational database and does not depend on the local schema or the underlying database management system. Nevertheless, the queries have to match the instances of the Relational.OWL ontology. For a detailed description on how to simulate the main operators of the relational algebra in RDQL and SPARQL, we again refer to [5] and [7].

3. DEMONSTRATION

The presentation of the RDQuery system consists of two main parts. We will first introduce the Java-based user interface of RDQuery, where the users can interactively query relational databases using RDQL and SPARQL, the RDF query languages currently implemented in the system. The GUI enables the users to follow the translation process, to verify the generated SQL query, and to regard the result set returned from the database quickly. Furthermore, the users can access their own query history and get a general idea of the tables stored in the corresponding database. We will start with the simulation of the basic relational algebra operators and get to more complex queries containing several join operations. Thereby we will describe the basic functionality of RDQuery and explain in-depth, how the queries are parsed and translated into SQL.

In the second part of the presentation we will demonstrate how Semantic Web applications can use the API of RDQuery to query and access information actually stored in relational databases, as if this data would actually be a part of the Semantic Web. Additionally, we will show how to create a mapping from the relational model to an arbitrary ontology simply using RDQuery and SPARQL. For this purpose we will create a SPARQL query, which actually maps the data stored in a typical relational database to instances of the 'Friend of a Friend' (FOAF) ontology. This data is then accessed by an application to perform several reasoning tasks, e.g. find people within the same social network, working on related projects, living in the same city, or listening to similar music. These reasoning tasks are all processed by the application without actually noticing, that the data is stored in and modeled for a relational database and not for the Semantic Web.

To illustrate the independency of the translation process from the concrete database schema and the underlying database management system, all queries presented in both parts of the presentation will be performed using several databases stored in different database systems.

4. REFERENCES

[1] JavaCC - Java Compiler Compiler. `https://javacc.dev.java.net/`, 2006.

[2] Jena - A Semantic Web Framework for Java. `http://jena.sourceforge.net/`, 2006.

[3] F. Manola and E. Miller. RDF primer. `http://www.w3.org/TR/2004/REC-rdf-primer-20040210/`, 2004. W3C Recommendation.

[4] D. L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview. `http://www.w3.org/TR/2004/REC-owl-features-20040210/`, 2004. W3C Recommendation.

[5] C. Pérez de Laborda and S. Conrad. Querying Relational Databases with RDQL. In *Berliner XML Tage*, pages 161–172, 2005.

[6] C. Pérez de Laborda and S. Conrad. Relational.OWL - A Data and Schema Representation Format Based on OWL. In *Conceptual Modelling 2005, Second Asia-Pacific Conference on Conceptual Modelling (APCCM2005), Newcastle, NSW, Australia, January/February 2005*, volume 43 of *CRPIT*, pages 89–96. Australian Computer Society, 2005.

[7] C. Pérez de Laborda and S. Conrad. Bringing Relational Data into the Semantic Web using SPARQL and Relational.OWL. In *Semantic Web and Databases, Third International Workshop, SWDB 2006, Proceedings of the 22nd International Conference on Data Engineering Workshops, ICDE 2006, 3-7 April 2006, Atlanta, GA, USA*. IEEE Computer Society, 2006.

[8] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. `http://www.w3.org/TR/2006/CR-rdf-sparql-query-20060406/`, 2006. W3C Candidate Recommendation.

[9] A. Seaborne. RDQL - A Query Language for RDF. `http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/`, 2004. W3C Member Submission.

Ontologies Change and Queries Break: Towards a Solution

Yaozhong Liang
Intelligence, Agents and Multimedia Group
School of Electronics and Computer Science
University of Southampton
Highfield, Southampton
England, United Kingdom
yl504r@ecs.soton.ac.uk

Harith Alani
Intelligence, Agents and Multimedia Group
School of Electronics and Computer Science
University of Southampton
Highfield, Southampton
England, United Kingdom
ha@ecs.soton.ac.uk

Nigel Shadbolt
Intelligence, Agents and Multimedia Group
School of Electronics and Computer Science
University of Southampton
Highfield, Southampton
England, United Kingdom
nrs@ecs.soton.ac.uk

ABSTRACT

Keeping track of ontology changes is becoming a critical issue for ontology-based applications. Updating an ontology that is in use may result in inconsistencies between the ontology and the knowledge base, dependent ontologies and applications/services. Current research concentrates on the creation of ontologies and how to manage ontology changes in terms of mapping ontology versions and keeping consistent with the instances. Very little work investigated controlling the impact on dependent applications/services; which is the aim of the system presented in this paper. The approach we propose is to make use of ontology change logs to analyse incoming RDQL queries and amend them as necessary. Revised queries can then be used to query the ontology and knowledge base as requested by the applications and services. We describe the design of our prototype system, and discuss related problems and future directions.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Abstracting methods; H.3.3 [**Information Search and Retrieval**]: Query formulation; I.2.4 [**Knowledge Representation Formalisms and Methods**]: Representation languages

General Terms

Ontology Management

Keywords

Ontology Change Management, Ontology Versioning, Knowledge Management, Semantic Web

1. INTRODUCTION AND RELATED WORK

Ontologies are quickly becoming indispensable parts of the Semantic Web. The number of ontologies that are being developed and used by various applications is continuously increasing. One of the major problems with ontologies is change. Ontology changes may cause serious problems to its data instantiations (the knowledge base), the applications and services that might be dependent on the ontology, as well as any ontologies that import that changed ontology [3].

Most work so far has focused on ways to handle ontology change, such as change characterisation [3], ontology evolution [4], ontology versioning [2], and consistency maintenance [5, 6, 7]. However, not much has been done with respect to using change-tracks to eliminate or reduce any impact that ontology change can have on any dependent applications and services. It would be very costly and perhaps even unrealistic to expect all parties that could be affected by a change to coordinate any such changes [1]. Therefore, we believe that it would be very beneficial to have a system that could track such changes, relate changes to incoming queries, amend such queries accordingly, and inform the query source of those changes and actions taken.

In this paper we describe a prototype system that targets these problems. The system uses a semantic log of ontology change to amend RDQL queries sent to the ontology as necessary. Such a system could save many hours of application re-development by not only updating queries automatically and maintaining the flow of knowledge to the applications as much as possible, but also to inform the developers of such changes in the ontology that relates to their queries.

2. SYSTEM DESCRIPTION

The solution shown in Figure 1 to tackle the identified problems is described as a series of steps as follows:

1. **Capture**: The changes made between two versions of the same ontology is captured at this stage. Currently, we identify changes by comparing two versions using PromptDiff in Protégé [4].
2. **Instantiate**: The *Log Ontology* is populated with change information identified in step 1.
3. **Analyse**: Queries submitted by the applications are analysed to find out whether any of the entities within the queries could be affected by the changes stored in the Log Ontology.

Posters and Demos of the 15th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2006 Podebrady, Czech Republic, 2nd-6th October, 2006

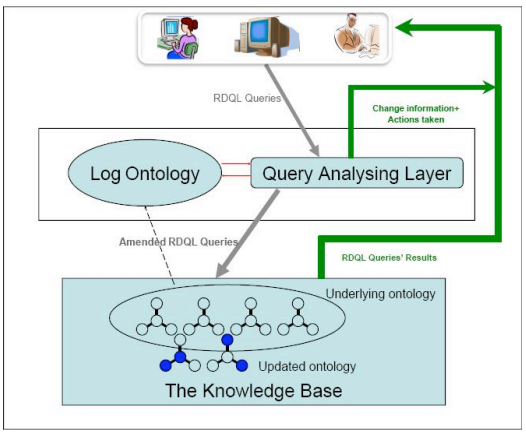


Figure 1: An overview of the Approach

4. **Update**: If entities within the queries are found to have been changed, they are replaced with their changes to form the new queries with updated entities, and then resubmitted to the queried ontology.

5. **Respond**: After the new-formed queries are submitted to the ontology for processing, the results are returned back to the application. At the same time, a summary of change/update information will also be returned back to the end-users with the query results so as to inform users of the updates.

Analyse, Update and Respond are implemented in the Middle Layer System in Figure 1. Its working process is presented in Figure 2.

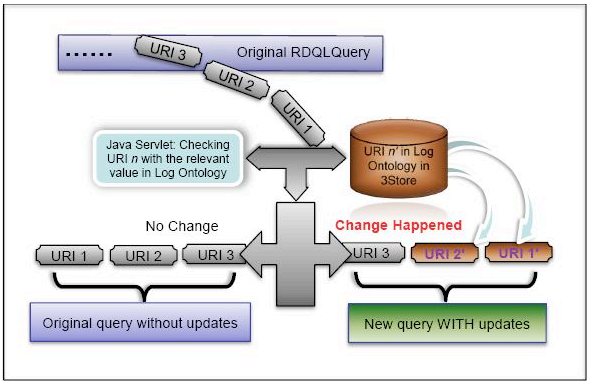


Figure 2: The working process of the Middle Layer System

3. CONCLUSIONS AND FUTURE WORK

We proposed an approach for handling ontology changes by means of using change-tracks to eliminate or reduce any impact that ontology change can have on the application queries. We developed a prototype system that analyses the incoming queries, amends the entities within the queries according to the change information stored in the Log Ontology built to store and manage change information between ontology versions, and informs the end-user of any changes and actions taken. We showed that with the extra support of the middle layer, some of the queries that are targeting parts of the ontology that have changed can be updated and processed properly.

In our next stage work, Enabling *Log Ontology* to capture a series of changes between multiple versions of the same ontology would be a necessity to assist our system to cope with more complex changes. In addition, (semi-)automatic collecting ontology change information between ontology versions would make our system usable in a large scale. Providing more machine-processable formats, such as RDF or OWL, of the query result would be beneficial for agents to understanding the change information within Semantic Web-based applications.

4. ACKNOWLEDGMENTS

This work has been supported under the Advanced Knowledge Technologies Interdisciplinary Research Collaboration (AKT IRC), which is sponsored by the UK Engineering and Physical Science Research Council under grant number GR/N15764/01. Special thanks for the excellent technical supports from my colleague David Dupplaw (dpd@ecs.soton.ac.uk).

5. REFERENCES

[1] Heflin, J. and Hendler, J. Dynamic ontologies on the web. In *Proceeding of the 17th American Association for Artificial Intelligence Conference (AAAI)*, pages 443–449, Menlo Park, CA, US, 2000. AAAI/MIT Press.

[2] Huang, Z. and Stuckenschmidt, H. Reasoning with multi-version ontologies: A temporal logic approach. In *Proceeding of the 4th International Semantic Web Conference (ISWC)*, Galway, Ireland, 2005.

[3] Klein, M. and Fensel, D. Ontology versioning on the semantic web. In *Proceeding of International Semantic Web Working Symposium (SWWS)*, Stanford University, California, U.S.A, 2001.

[4] N. K. Klein, M., and Musen, M.A. Tracking changes during ontology evolution. In *Proceeding of the 3rd International Semantic Web Conference (ISWC2004)*, Hiroshima, Japan, November 2004.

[5] Noy, N.F., and Musen, M.A. Promptdiff: A fixed-point algorithm for comparing ontology versions. In *Proceeding of the 18th National Conference of Artificial Intelligence (AAAI)*, pages 744–750, Edmonton, Alberta, Canada, 2002.

[6] K. K. Ognyanov, D., and Fensel, D. Ontology versioning and change detection on the web. In *Proceeding of 13th International Conference on Knowledge Engineering and Management*, Siguenza, Spain, 2002.

[7] H. H. H. Stuckenschmidt, H., and Sure, Y. A framework for handling inconsistency in changing ontologies. In *Proceeding of the 4th International Semantic Web Conference (ISWC)*, Galway, Ireland, 2005.

Activity-Theoretical Model as a Tool for Clinical Decision-Support Development

Helena Lindgren
Department of Computing Science, Umeå University
SE-901 87 Umeå
Sweden
helena@cs.umu.se

ABSTRACT
Clinical investigation activities are complex processes, which are situated, emergent and directed by the individual need of the patient, but also restricted or enhanced by the available resources at different points in the process. For the purpose of creating a system which provides support throughout the investigation process, i.e. functioning as a cognitive tool for the user, the clinical investigation process needs to be assessed and formalized. The presented work is based on analyzes of the domain knowledge and case studies of investigations of actual patients and provides a conceptual model for a clinical investigation activity. The framework of cultural-historical activity theory was used for the interpretation of the data and the model is used for identifying which actions are appropriate for formalization in a decision-support system.

Categories and Subject Descriptors
I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems—*Medicine and science*; H.4.2 [**Information Systems Applications**]: Types of Systems—*decision support*; H.1.2 [**Models and Principles**]: User/Machine Systems—*Human factors*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

General Terms
Theory, Human Factors, Design

Keywords
activity theory, knowledge representation

1. INTRODUCTION
In the development of a decision-support system for the clinical investigation of dementia [3], the investigation process was analyzed in terms of activity theory in order to identify structures suitable for formalization. The extended activity system described by Engeström [2] was used to identify the basic building-blocks of an activity which are formalized. The main activity, investigation of dementia, is defined by its motive; the altered and improved situation of the patient as the anticipated outcome. For each sub-action in the process an activity system can be defined, as well as for the whole activity. The outcome of each action is another piece of evidence, related to the patient. Consequently, typically each action concerns a change of the incomplete knowledge about the patient's situation. The actions, or processes, are oriented in time and dependent on the activity system to accomplish the change, the outcome. It is important to take the patient-oriented view of the activity, since although it is the same type of activity for two different patients, the execution will most likely differ, due to different needs of the patients, available resources, which tools are used, and decisions made during the process.

2. THE MODEL
In terms of activity theory the activity system in focus includes the basic and originally entities the actor (subject), the object and tools, which are entities with certain characteristic properties and roles in the activity system. Their relations can be summarized by viewing the actor (medical professional) changing the patient's life situation (object and focus for activity) by using tools. The tools mediate the activity and should not be in focus for the activity. Entities such as the patient's life situation, or the knowledge regarding the patient, change because they instantiate different properties at different times. As a bi-product of the activity, the actor necessarily changes as well in the process, by gaining new skills and knowledge. The original model was extended with the context of activity, including rules for and division of work, in [2]. The contextual factors can be seen as properties of or relations between social entities such as individual actors, teams or organizations. This structure of contextual factors includes values and priorities these entities hold. We will summarize the most important components of the activity for our purposes which are actions, object and tools.

The process of investigating cognitive diseases involves different kind of actions, which can be lower-level actions such as executing a blood test but also actions of analytic and decisive nature which is typically performed in the mind of the actor. These types are commonly distinguished as ontic actions (aimed at changing the environment) and epistemic actions (aimed at changing knowledge states of an agent).

Activity theory defines three levels of human actions; activity, consisting of a set of actions, which in turn may consist of actions and operations in a nested structure. An activity is defined by its motive and constitutes the minimal unit of analysis of purposeful activity. Actions have goals and are executed by the actor at a conscious level, in contrast to operations which do not have a goal on their own and which are executed at the lowest level as automated, unconscious processes. The structure is dynamic in that there may be a frequent transformation between the levels, triggered by the demands and prerequisites in the environment or factors in the actor such as lack of knowledge. In our work the levels of activity is used to identify the levels of actions which typically correspond to the experienced actor in the investigation process and to distinguish between tasks of different complexity. We distinguish between different types of actions by the purpose, or the goal of a particular action. The following types were identified when the investigation process was analyzed:

- IO, investigation actions (typically activities),
- CO, object-(data-) creating actions (typically operations),
- AO, object- (data-) analyzing/transforming/refining actions,
- DEO, actions which aim at determine the (amount of) existence of objects,
- DTO, actions which determine the type of object (e.g. differential-diagnosis),
- CHO, object-changing actions (include ontic actions such as interventions),
- ECHO, object-change evaluation actions, necessarily included in CHO actions.

A general definition of the complex sort action can be the following:

$$action : [Object, Goal, C, Actor, A, T, Outcome] \rightarrow Action$$

where action is the constructor, Goal defines the sort and purpose of the action, C represents the context, which is the set of related formal or informal social organizations or persons involved and the rules and division of labor governing the behavior of its members, and Actor is one or several subjects representing the social organization responsible for the execution of the activity, A is a set of actions, T is a set of tools. The Outcome is simply the changed Object.

The object in focus for activity can be an abstract mental construct such as medical knowledge or a physical entity such as the human body and its parts. The purpose of focussing the object in an activity is to change it, therefore each object has measurable or describable qualities which are in focus in the activity. The constructor for the sort Object becomes:

$$object : [Entity, Qualifier] \rightarrow Object$$

The object and its qualities is the current state of the object at a certain time point, for instance, when the action is initiated, and the outcome of an action (i.e. the Object and its qualities after the execution), is defined as the Evidence, which is used in subsequent reasoning. The outcome is interpreted into evidence by the constructor:

$$evidence : [Outcome, Tool] \rightarrow Evidence$$

The tool is a key entity of the activity system when the reasoning process is to be formalized. The notion of tool in the perspective of activity theory can be physical entities or mental constructs such as certain knowledge or models of reality. It is the activity-mediating role a particular entity holds in an activity that defines it as a tool. The constructor for a tool in our model becomes

$$tool : [Entity, Q] \rightarrow Tool$$

where Q is the substance of the tool which is to be used, including directives of how the tool is to be used. For instance, if the entity is a clinical guideline, then Q represents its medical and procedural knowledge, the content of scales, etc. In the perspective of clinical reasoning, the tool can constitute a part of or a set of clinical guidelines that is used in an action. As an example from the domain of dementia, a knowledge tool such as the clinical guideline DSM-IV-TR [1] may come equipped with a formalization of the content in the form of a set of sentences as part of a logic language:

$$tool : [DSM - IV, \Phi_{\mathcal{L}^\pi}^{DSM-IV}] \rightarrow Tool$$

where $\Phi_{\mathcal{L}^\pi}^{DSM-IV}$ consist of a set of rules formulated in propositional logic $\mathcal{L}^\pi$, which correspond to sets of features necessarily present or absent in a patient in order to establish the type of dementia formulated in the guideline [4]. Consequently, it is the rules of Q that produces the Qualifier of the Object in focus.

3. CONCLUSIONS

The presented components of activity in the perspective of cultural-historical activity theory are useful for framing general as well as specific aspects of clinical activity. The model of the activity in focus, created using the constructors described in this work, serves as a tool for identifying tasks and components in the reasoning process which are to be formalized and supported by a decision-support system for the clinical practice. It also gives design implications for the interaction with the system.

4. REFERENCES

[1] American Psychiatric Association. Diagnostic and statistical manual of mental disorders, fourth edition, text revision (DSM-IV-TR). American Psychiatric Association, 1994.

[2] Y. Engeström. *Learning by expanding: an activity-theoretical approach to developmental research.* Orienta-Konsultit, Helsinki, 1987.

[3] H. Lindgren. Managing knowledge in the development of a decision-support system for the investigation of dementia. Licentiate thesis, UMINF 01/05, Department of Computing Science, Umeå University, Sweden, 2005.

[4] H. Lindgren and P. Eklund. Differential diagnosis of dementia in an argumentation framework. *Journal of Intelligent and Fuzzy Systems*, 16:1–8, 2005.

Managing R&D European Projects with ODESeW

Asunción Gómez-Pérez, Angel López-Cima
M. Carmen Suárez-Figueroa
Universidad Politecnica de Madrid.
Facultad de Informática.
Campus de Montegancedo, s/n.
28660 Boadilla del Monte, Madrid, Spain
+34913367467
{asun,alopez,mcsuarez@fi.upm.es}

Oscar Corcho
University of Manchester
School of Computer Science
Oxford Road, Manchester, United Kingdom
+44(0)1612756821
Oscar.Corcho@manchester.ac.uk

ABSTRACT
This demo presents a Semantic-Web-based knowledge management system that supports R&D European Projects in different aspects: dissemination of project information and generation of management reports for the European Commission (EC).

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval]: Information Search and Retrieval

General Terms: Design

Keywords: Semantic Web, framework, Web application.

1. INTRODUCTION
In the field of Web application engineering, a large amount of content management systems (CMSs) are available for the development of standard Web applications. Among them we can cite the following: Zope[1], Mambo[2], Ruby on Rails[3], etc. All of them allow developers to generate in short time a portal to publish their information and, with some extra effort, to implement application-specific functionalities. These CMSs are oriented towards presenting information to human users, not to other software systems, and very few of them allowchanging the data model used in the portal.

ODESeW [3] is an application development framework, based on Semantic Web technologies, which overcomes the two previous limitations. In this demo we show how it can be used to implement a CMS for R&D European Projects. Using this CMS, members of the organisations involved in the project can manage all types of information about their organisations, persons, project meetings, all sorts of documents and deliverables, progress and administrative reporting information, etc. Furthermore, the project coordinator can manage the project progress using different types of reports that can be easily maintained. Besides external users can access the HTML pages generated by the CMS, taking into account the permissions in the system, and other software agents can access this information in other formats like RDF, RDF Schema and OWL.

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

[2] http://www.mamboserver.com/

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

2. ODESeW
ODESeW (Semantic Web Portal based on WebODE) was first described in [4] as a tool that could be used for the automatic generation of Web portals where all the information was indexed by means of ontologies. This portal generation system was built on top of the WebODE ontology engineering workbench [1], thus inheriting many of its features, such as the deployment of ontologies in databases, the availability of import and export functions from and to different ontology languages, etc.ODESeW has now evolved into a more comprehensive application development framework that eases the maintainability and personalisation of the content generated, while maintaining the feature of automatic application generation.

One of the main innovative features of ODESeW is the navigation and composition model [3]. This model allows Web developers to specify explicity how users will navigate the application and also allows them to reuse views more easily.

In ODESeW, views can be designed using JSTL [1] and JavaBeans [2]. They allow creating highly reusable views that can tolerate changes in the data model or to create fit views for specific information in the ontologies (concepts, instances) that are vulnerable through ontology changes.

Besides these content provision, visualization, and access functions, ODESeW provides more functionalities like a search engine, content implementation in different languages (RDF, RDFS and OWL) and administrator functionalities for user management, read/write permission management and selection of ontologies to be used in the portal.

ODESeW can manage different domain ontologies, which can have relations between themselves. Besides, it represents application users by means of an application-independent User Ontology. This ontology stores the different user profiles of the portal and has only two main concepts: User and Role. The User Ontology can be extended in the different web applications by adding attributes or relationships to any of the application-specific domain ontologies. In this way, the User Ontology can link a user to another piece of information in the portal, for example, an organization.

3. Ontologies in a R&D European Project
To describe a collaborative project we have used the following six ontologies, which can be easily reused for describing other similar

projects[4]: the **Documentation Ontology** models knowledge of documentation used in the project; the **Event Ontology** models knowledge of events that are related to the project; the **Organization Ontology** models knowledge of organizations that work in the project; **Person Ontology** models knowledge of persons who work in the project; the **Project Ontology** models the Technical Annex of a project, including information about: milestones, workpackages, tasks, projects or networks of excellence, etc.; the **Management Ontology** models the periodic reports that the consortium of the project must sent to the EC.

4. A case study of a R&D portal: Knowledge Web

The functionalities that are provided by the Knowledge Web portal are divided according to the different types of users that can access it. In the case of Knowledge Web portal there are two general users (partner and administrator user) to assert information in the ontologies that are publish in the portal as a part of public information and three user (reporting, area manager and project coordinator users) focusing to generate management document that are required by the EC.

Partner User

This general user is responsible of inserting his/her organization information and the information of all the participants in the project from his/her organization. If the partner is a workpackage leader, he/she is also responsible to upload deliverables inside the portal. Besides, an user of this type can insert concrete meetings, conferences, workshops, etc.

Administrator

This user is in charge of creating new users, setting their read and write permissions and specifying which ontologies in the ontology server (WebODE) are being managed inside the portal. The administrator is also allowed to change the ontologies inside the ontology server.

Besides all administration issues, this user is in charge of including all the project definition information: workpackages, deliverables, global efforts of each partner, etc.

Reporting User

When a reporting user logs into the system and goes into the reporting section, the portal shows all the tasks to be done. These tasks are:

- **Workpackage progress reports**, for each workpackage that the user's organisation is leader of.
- **Effort report** for the organisation to which the user belongs.

Area Manager

In a large project, workpackages can be organised in different areas (this is specified in the project ontology). In the context of Knowledge Web there are four areas: industrial, research, educational and management. Each area has several workpackages associated and has also a person that is responsible of it, known as the area leader. In the **activity report**, area managers can include an area overview about the general progress of the area.

Managing Director

The managing director is a person that belongs to the project coordinator organization and is in charge of monitoring the progress of all reports generated by individual partners and generates and downloads a draft version of the **activity report**.

When this user logs into the system and accesses the reporting system, the portal shows the effort reports from all the project partners and the progress reports from all the workpackages. Besides, there is a link to a view for monitoring the current status of all the reports

The **activity report** is one of the documents that must be delivered by the project coordinator to the European Commission. This document compiles all the **workpackage progress reports**, the **effort reports** and the **area overviews** in one document. The generated document is presented in HTML and in MS Word formats. This document is a draft version in which the Managing Director can modify with specific information that only the project coordinator can include.

4.1 Others Knowledge Web functionalities

A part of the different users and differents views and forms for each of them, ODESeW gives other functionalities. These functionalities are the messenger service and mailing system.

The **messenger service** sends events from the portal like the request of a view from a user, the logging event of a user, the editing of an instance, a schedule event, etc. These events are sent to the messenger service and this one redirect all these events to others applications.

The **mailing service** generates a dynamic mailing using the domain ontologies.

These services are connected in Knowledge Web using the mailing service as a receipt of some events from the messenger service. In this way, the portal notify to the administrator when a progress report is submitted and send a warning message to the partner, area managers and the project administrator which reports are delayed according a schedule.

5. ACKNOWLEDGMENTS

This work has been supported by the EU IST Network of Excellence Knowledge Web[5].

6. REFERENCES

[1] Delisle, P..*A Standard Tag Library for JavaServer Pages*. Sun Microsystems. JSR-000052. Noviembre 2003.

[2] Hamilton, G. *JavaBean*. Sun Microsystems. August 1997.

[3] Corcho, O; Lopez-Cima, A.; Gómez-Pérez, A. *A platform for the development of Semantic Web portals*. Internation Conference on Web Engineering. ICWE 2006. July 2006.

[4] Arpírez JC, Corcho O, Fernández-López M, Gómez-Pérez A. *WebODE in a nutshell*. AI Magazine 24(3):37-48. Fall 2003.

[4] In fact, they have been already used in four EU projects of diifferent nature (Esperonto, OntoGrid, Knowledge Web and NeOn).

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

HOLA: A Hybrid Ontology Learning Architecture

David Manzano-Macho and Asunción Gómez-Pérez
Facultad de Informática
Universidad Politecnica de Madrid
Campus de Montegancedo, sn
28660 Boadilla del Monte, Spain
{dmanzano,asun}@fi.upm.es

Daniel Borrajo
Departamento de Informática
Universidad Carlos III de Madrid
Avda. de la Universidad, 30
28911 Leganes Spain
dborrajo@ia.uc3m.es

ABSTRACT

In this paper, we describe a hybrid, extensible and domain-independent ontology learning architecture (HOLA). It is based on the combination of different types of both information sources and techniques, with minimum user intervention. HOLA aims to achieve two main goals: semi-automatic enrichment of existing domain ontologies; and learning of how to do it better. We focus in this paper in the first goal.

Keywords

Ontology Learning, Knowledge acquisition

1. INTRODUCTION

The amount of data generated by the success of Internet is demanding methodologies and tools to automatically extract unknown and potentially useful knowledge out of it, and generating structured representations with that knowledge. The research on ontology learning has made possible the development of several approaches that allow the partial automation of the ontology construction process [1]. Therefore, most of the state of the art approaches require an intensive user intervention to achieve the learning process. Besides, none of them is able to learn how to improve the internal process followed to enrich a domain ontology, and to choose the most suitable combination of available techniques to solve the acquisition problem. They assist on learning some parts of an ontology but they are not able to learn how to do the acquisition process better. In this context, and as a partial solution of these problems, we present an open, flexible, easily extensible, domain-independent, and scalable architecture for learning ontologies (HOLA).

2. HYBRID ONTOLOGY LEARNING ARCHITECTURE

HOLA aims to reach two main goals: to assist on learning a new ontology or improving an existing one, and to learn how to better build new ontologies, by means of the introduction of an *iterative feedback* into the system, all done with *minimum user intervention*. This feedback changes the way in which the different components of the architecture are selected to build an specific ontology according to the provided sources, and how to combine their results. This paper is focused, for the sake of brevity, to present just how HOLA assists on enriching an existing ontology. As shown in Figure 1, the architecture is composed of five main phases: *processing, acquisition, action, consolidation,* and *evaluation*.

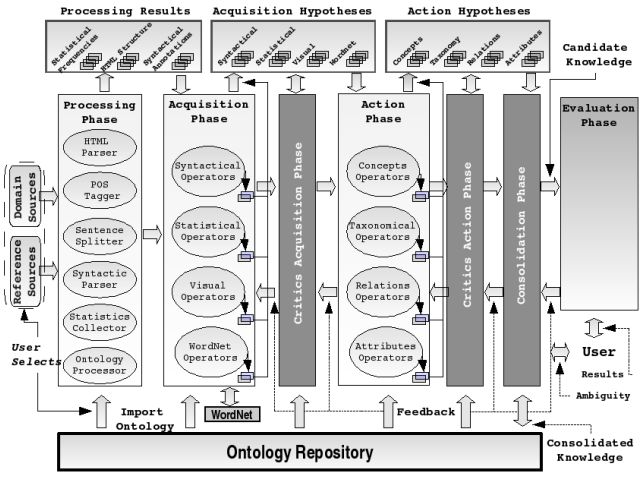


Figure 1: Design of the Architecture for Learning Ontologies.

The architecture has a *modular design*, composed by a set of consecutive phases covering all the steps of the ontology construction process. This design facilitates the extensibility and reusability of the components included in each phase. Besides, each phase is affected by the evaluation of its results made by the subsequent phases. The architecture is *hybrid* in the sense that it uses different types of information sources, and provides an effective combination of different methods for extracting and analyzing that information. HOLA is based on the use of a set of *operators* that conform each phase. An operator, in this context, is a generic knowledge transformation element with preconditions (conditions on the available information at each phase) and effects (cre-

EKAW 2006 - 15th International Conference on Knowledge Engineering and Knowledge Management Managing Knowledge in a World of Networks
2nd-6th October 2006 - Podebrady, Czech Republic

ation, modification or deletion of hypothesis). The operators are responsible of processing the selected inputs, extracting relevant information out of them, performing the construction of new ontologies, and evaluating the final results. This evaluation produces a *feedback* that will change what operators are selected, and how their outcomes are combined to obtain better results. Each of the operators analyses and combines the hypotheses generated by the previous phase and produces new ones after executing their actions. The *hypotheses* are statements about what HOLA believes to be true about an element at a given moment: its type, visual place and visual relations with other elements, as well as syntactical and statistical information. The hypotheses could be wrong, and HOLA has the ability of deciding whether the generated hypotheses go towards the next phase or they are rejected. Besides, some hypothesis could reinforce others providing additional source of evidence about their correctness and suitability. This is the role of the *critics phases* that are executed after the acquisition and action phases, with the objective of evaluating the produced hypothesis, and to decide whether they have to be accepted or rejected. Now, the goals of each phase are briefly explained.

- The process starts with the *sources selection* made by the user. The user has to provide domain sources; a set of documents that sufficiently describe the domain of the ontology. The user might optionally provide other complementary sources, called reference sources, that are not specific to any domain. The *reference source* will be used as an indicator of the relevance of a term to the domain, comparing the frequency of appearance of a term in the domain and reference sources, following an statistical approach. Finally, HOLA also accepts an existing domain ontology that will be enriched as a result of the learning process.

- *Processing Phase*: transforms the selected sources into an internal model manageable by the system. This model stores the syntactical, statistical, and visual information produced as a result of the processing phase. Depending on the type of sources provided by the user, the artifacts that have to be used to perform the processing can vary. In our first experiments, the sources are web pages. Therefore, HOLA offers operators to process each document by its text, visual layout, and statistical frequency of appearance for every term in the document.

- *Acquisition Phase*: extracts candidate ontological elements, and relations among them, from the processed information. It has access to other available resources like WordNet and the selected domain ontology. Therefore, the type of hypotheses that the system generates at this phase relates elements to their syntactic characteristics, statistical measures, visual position, and characteristics within the document. In the case of including new types of sources, it is only necessary to include new operators that understand the content of the new sources and the acquisition hypothesis produced using them. Once the acquisition phase finishes, the resulting hypotheses pass through a *critic phase*. Here critic means a filter that aims to ensure: the suitability of the hypothesis; that they do not contain contradictions among different hypothesis; or they are relevant enough for the ontology domain.

- *Action Phase*: transforms the acquisition hypotheses into action hypotheses: what to do with the document elements in ontological terms. After finishing this phase, the action hypothesis about new ontological elements are evaluated by a *second critic phase* to ensure their quality and suitability.

- *Consolidation Phase*: performs the actions received from the previous phase, augmenting the domain ontology using the new candidate knowledge. The hypotheses are presented to the user in case of ambiguity who will decide if the hypothesis is correct.

- *Evaluation Phase*: analyses the obtained results, considering the decisions made by the user during the consolidation phase, and the finally accepted hypotheses. Using the results of this analysis, a feedback to the previous phases is produced, aiming to improve the results in future uses. This step has not been implemented yet.

3. CONCLUSIONS

The open, extensible, and domain-independent architecture for learning ontologies (HOLA) presented in this paper, combines different types of both sources and techniques to improve the detection of potential useful knowledge. It is based on the analysis of different sources of evidence that tell how relevant an element is from the point of view of the target domain. HOLA covers all phases, from the processing of the selected sources to the final evaluation of the results. Each phase is composed of several operators that produce hypotheses about the new elements found in the selected inputs. These hypotheses are statements about new acquired elements and their relation with other elements. The hypotheses enter critics phases that ensure their correctness and validity. These critics modify which of the operators are selected considering the evaluation of their produced hypotheses. The combination of the different hypotheses, produced using different sources and by the application of several techniques, constitutes a reinforcement about the correctness of the hypotheses. The final analysis of the whole process, the interaction with the user, and the new knowledge introduced in the ontology, produce a feedback to the previous phases for future uses.

We have completed the implementation of the first prototype of HOLA, including the processing modules for Web sources. This prototype has available all the acquisition and action operators mentioned previously, and a set of critics that evaluates the hypotheses generated during the process.

4. ACKNOWLEDGEMENTS

This work has been partially supported by a research fellowship from Formación de Personal Investigador of the Comunidad de Madrid, Spain, MEC project TIN2005-08945-C06-05 and regional CAM-UC3M project UC3M-INF-05-016.

5. REFERENCES

[1] A. Gómez-Pérez and D. Manzano-Macho. An overview of methods and tools for ontology learning from texts. *Knowledge Engineering Review*, 19:187–212, 2005.

DBin – enabling SW P2P communities

C.Morbidoni, G.Tummarello, 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, multi-platform Semantic Web application 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 metaphor 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.

Categories and Subject Descriptors

H.4.3 Communications Applications

General Terms

Algorithms, Management, Experimentation.

Keywords

Semantic Web, RDF, Ontology, User interface, p2p.

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,

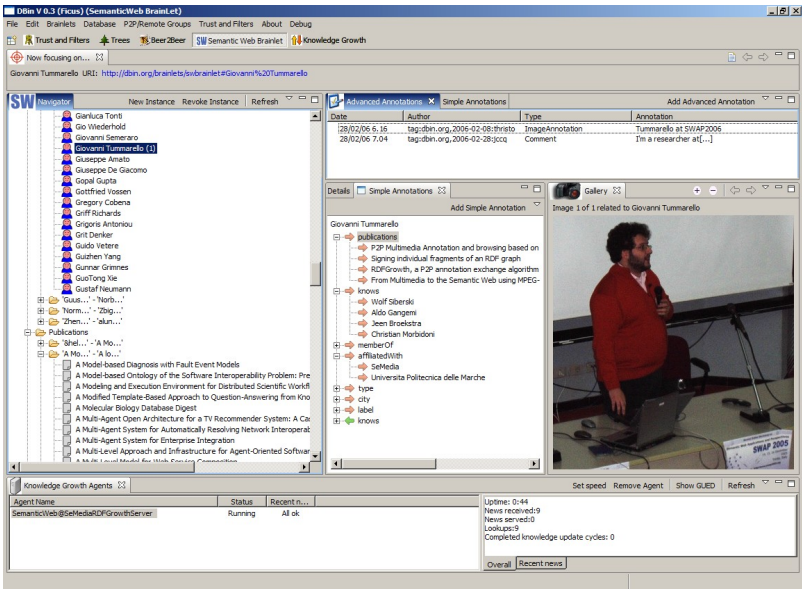


Figure 1 A screen shot of the SemanticWeb research Brainlet running. The principal "views" are: an ontology (and instances) browsing Navigator, a set of "Annotation" views and others related to searching, browsing, filtering etc. Kickstarting data is delivered insider the brainlet itself and has been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

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.THE RDFGROWTH ALGORITHM

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 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 THE 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.CONCLUSION AND FUTURE WORK

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

Similarity Mapping with Uncertainty for Knowledge Management of Heterogeneous Scientific Databases in a Distributed Ontology-Mapping Framework

Miklos Nagy
Knowledge Media Institute (KMi)
The Open University
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
mn2336@student.open.ac.uk

Maria Vargas-Vera
Knowledge Media Institute (KMi)
The Open University
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
m.vargas-vera@open.ac.uk

Enrico Motta
Knowledge Media Institute (KMi)
The Open University
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
e.motta@open.ac.uk

ABSTRACT
This paper describes a framework for integrating similarity measures and Dempster-Shafer belief functions for knowledge management of heterogeneous scientific databases in the context of multi agent ontology mapping. In order to incorporate uncertainty inherent to the ontology mapping process, we propose utilizing the Dempster-Shafer model for dealing with incomplete and uncertain information produced during the mapping. A novel approach is presented how assessing belief can influence the similarities originally created by both syntactic and semantic similarity algorithms. Our approach is an alternative to the classical Bayesian reasoning which has been investigated for improving the efficiency of creating ontology mappings.

Categories and Subject Descriptors
I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents, Languages and structures, Multiagent systems.

General Terms
Algorithms, Performance, Design, Languages, Theory.

Keywords
Ontology Integration, Question answering , Uncertainty and knowledge modelling.

1. INTRODUCTION
With the continuously increasing amount of data produced by electronic systems the integration of data and knowledge from multiple heterogeneous sources is an important problem that Scientific Database Community is facing today. To share information different solutions have been proposed that utilize distributed ontologies and ontology mapping as a source of the common knowledge. These architectures are the alternative to the federated approach, which is used to allow scientists to maintain control of their data, while sharing it within the community. An important aspect of ontology mapping in the context of knowledge management of heterogeneous scientific databases is how the incomplete and uncertain results of the different similarity algorithms can be interpreted during the mapping process. As the domains becomes larger and more complex, open, and distributed, a set of cooperating agents is needed to address the reasoning task effectively. In this context each agent carries only a partial knowledge representation about the domain and can observe the domain from a partial perspective where available prior knowledge is generally uncertain. Our novel approach utilizes a multi agent framework where different mapping agents provide similarity measures about particular entities (e.g. material, specimen, etc.) and uncertainty plays a central role interpreting such similarities. Our system considers query answering over Web enabled S&T (Scientific and Technical) or engineering databases which are described with their own domain specific ontologies.

2. ONTOLOGY MAPPING IN A MULTI AGENT SYSTEM
To achieve the necessary performance for a real time mapping we utilize multi agent architecture. Without the multi agent architecture the response time of the system can increase exponentially when the number of concepts to map increases due to the Dempster's rule of combination. The high-level system architecture (figure 1) shows how the functional parts of the system are related with each other.

1. Data: On the data layer the heterogeneous data sources are represented by their ontologies.
2. Mediator: In the mediator layer the agents are organized in different levels. Agents at the broker level are responsible for decomposing the query into sub queries, based on the meta-descriptors. The meta-descriptor is the key component of the system that describes what kind of information can be found in the different sources. Agents communicate through the blackboard which is a task independent architecture for integrating multiple knowledge sources e.g. different local agents. The blackboard holds the state of the problem solution, while the knowledge sources make modifications to the blackboard when appropriate.

3. User interaction: The AQUA query answering system itself, which provides precise answers to specific questions raised by the user.

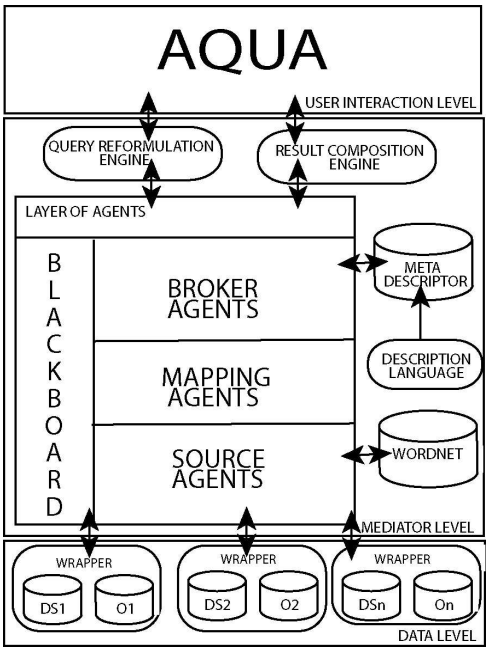


Figure 1. High level system architecture.

3. SIMILARITY

3.1 Syntactic similarity

To assess syntactic similarity between ontology entities we use different string-based techniques to match names and name descriptions. These distance functions map a pair of strings to a real number, which indicates a qualitative similarity between the strings. To achieve more reliable assessment we combine different string matching techniques such as edit distance like functions e.g. Monger-Elkan[1] to the token-based distance functions e.g. Jaccard[2] similarity. To combine different similarity measures we use Dempster's rule of combination (see section 4). At this stage of the similarity mapping our algorithm takes one entity from Ontology 1 and tries to find similar entity in extended query. The similarity mapping process is carried out on concept-names and property sets. The use of string distances described here is the first step in identifying matching entities between query and the ontology or between ontologies with little prior knowledge, or ill structured data. However, string similarity alone is not sufficient to capture the subtle differences between classes with similar names but different meanings. So we work with WordNet in order to exploit synonymy at the lexical-level.

3.2 Semantic similarity

For semantic similarity between concept, relations and the properties we use graph based techniques. We take the extended query and the ontology input as labeled graphs. The semantic matching is viewed as graph-like structures containing terms and their inter-relationships. The similarity comparison between a pair of nodes from two ontologies is based on the analysis of their positions within the graphs. Our assumption is that if two nodes from two ontologies are similar, their neighbours might also be somehow similar. We consider semantic similarity between nodes of the graphs based on similarity of leaf nodes. That is, two non-leaf schema elements are semantically similar if their leaf sets are highly similar, even if their immediate children are not. Assessing the above-mentioned similarities in our multi agent framework we adapted and extended the SimilarityBase and SimilarityTop algorithms [3,4] used in the current AQUA system for multiple ontologies.

4. UNCERTAINTY

In our framework we use the Dempster-Shafer[5] theory of evidence, which provides a mechanism for modeling and reasoning uncertain information in a numerical way particularly when it is not possible to assign a belief to a single element of a set of values. The main advantage of the Dempster-Shafer theory over the classical probabilistic theories is the evidence of different levels of abstraction can be represented in a way, which allows clear discrimination to be made between uncertainty and ignorance. Further advantage is that the theory provides a method for combining the effect of different learned evidences to establish a new belief by using Dempster's combination rule. An important aspect of the mapping is how one can make a decision over how different similarity measures can be combined and which nodes should be retained as best possible candidates for the match. Our algorithm takes all the concepts and its properties from the different ontologies and assesses similarity with all the concepts and properties in the query graph. To obtain more reliable results we need to combine the similarity assessments that have been produced by the different similarity algorithms. Our approach is to consider these measures as subjective probabilities and utilize a well-established framework that provides convenient way to represent and combine these probabilities.

5. REFERENCES

[1] Monge A. E., Elkan C. P. The field-matching problem: algorithm and applications. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, US,1996.

[2] Cohen W., Ravikumar P. and Fienberg S. A Comparison of String Distance Metrics for Name-Matching Tasks, In Proceedings of Information Integration on the Web (IIWeb 2003), Accapulco, Mexico, 2003.

[3] Vargas-Vera M. and Motta E. A Knowledge-Based Approach to Ontologies Data Integration. KMi-TR-152, The Open University, UK, 2004.

[4] Vargas-Vera M.. and Motta E. An Ontology-driven Similarity Algorithm. KMI-TR- 151, Knowledge Media Institute, The Open University, UK, 2004.

[5] Shafer G. A Mathematical Theory of Evidence. Princeton University Press,1976.

Refining search queries using WordNet glosses

Jan Nemrava
Department of Information and Knowledge Engineering
University of Economics Prague
W.Churchill Sq. 4, 130 67 Praha 3, Czech Republic

nemrava@vse.cz

ABSTRACT
This paper describes one of the approaches how to overcome some major limitations of current fulltext search engines. It tries to discover semantic categories for proper nouns from WordNet glosses and verify them with Yahoo and Google. It relies on lexical patterns present in large repositories and it is inspired by Pankow [2]. It follows the paper from student workshop in April 2006 [6]. Some limitations were solved and new results and plans are presented here.

Keywords
WordNet glosses, Search Engines, Hearst Patterns

1. INTRODUCTION
Fulltext search engines have recently become a basic tool for acquiring arbitrary information from the World Wide Web. Nevertheless, there still exist some limitations that play an important role in searching information within a keyword based search interface. One of the keyword-based search major problem is that people tend to insert too general queries (according to Search Engine Journal [1], in 2004 more than 50% of all queries inserted were one or two words long), which leads to a huge amount of returned hits to a given query. The way how to deal with a huge amount of returned web pages is to arrange the results according to their meaning using their semantic category. The purpose of this paper is to propose a technique how to refine the queries inserted into search engines using WordNet for discovering the synonyms and Hearst Patterns for discovering is-a relation between the queried term and its possible superclass (i.e. hypernym) concept, and to extend the previous work by using Google proximity search [5] and two fulltext APIs to speed up the queries.

2. SOURCES OF INFORMATION
The idea is to combine freely available information resources with several techniques to exploit the redundancy present within web sites. The following part describes each of them very briefly before describing the procedure and tests.

2.1 WordNet
The first precondition is that the given word is contained in WordNet [3]. WordNet glosses (one or two sentences long description of the concept) are acquired and processed with part-of-speech tagger; only nouns are retained.

2.2 Hearst Patterns
Hearst patterns are lexico-syntactic patterns that indicate the existence of class/subclass relation in unstructured data source, e.g. web pages. Examples of lexico-syntactic patterns that were described in [4] are following:

- NP_0 *such as* $NP_1, NP_2, \ldots, NP_{n-1}$ *(and | or)* NP_n
- *such* NP_0 *as* $NP_1, NP_2, \ldots, NP_{n-1}$ *(and | or)* NP_n
- $NP_1, NP_2, \ldots, NP_{n-1}$ *(and | or) other* NP_0
- NP_0 *(incl. —esp.)* $NP_1, NP_2, \ldots, NP_{n-1}$ *(and | or)* NP_n
- *and very common "*NP_i *is a* NP_0*"*

Hearst firstly noticed that from patterns above we can derive for all NP_i, $1 < i < n$ a hyponym (NP_i, NP_0). Given two terms t_1 and t_2 we are able *to record how many times* some of these patterns indicate an „is-a" relation between them. Although these patterns occur quite rarely in unstructured data, they provide reliable and valuable information.

2.3 Fulltext Search Engines API
Both Yahoo and Google provide API (Application programming interface) to access their databases. Both have limited queries per day (1000 in Google and approx. 5000 in Yahoo) and both provide the same services as the web-based interface. Yahoo's is much faster while Google accepts * as a wildcard.

2.4 NLP
Simple NLP (natural language processing) methods are applied to discover [8] and stem [7] nouns in WordNet glosses. Dictionary is used to eliminate stop-words.

2.5 Proximity search in Google
Kostoff et al. [1] described the way how to make Google answer the queries for words that are within a specified distance from each other.

3. GETTING HYPERNYMS
When a user inserts his query, which is assumed to be a proper noun in this state of work, the relevant WordNet synset is looked up and all meanings of the given word are obtained together with their glosses. These glosses are preprocessed by replacing stop-words according to the given dictionary, the part-of-speech tags are created with POS tagger [8] and the nouns are lemmatized and stored as a potential hypernym for the concept. These nouns are called the *candidate nouns.*

What follows is applying one of the Hearst patterns to discover which of the candidate nouns is a hypernym describing the given

proper noun. At first, the most common pattern *"NP_i is a NP_0"* is used to query search engine. The word given by a user is considered as NP_i and each candidate noun is tested with this pattern as NP_0. Only the numbers of results are counted. To verify the information provided by the *"is-a pattern"* we employ another pattern as a verification It I is *"NP_0 and other NP_1s"* (NP_1 must be in plural). The procedure is repeated for each candidate noun and only numbers of results are kept. The values are normalized and compared. The one with the highest value is considered to be a *hypernym* for the given concept.

4. EXAMPLE

The following example shows how to discover syntactic meanings of the word "Pluto". At the end the main limitations are stated.

WordNet glosses for concept Pluto

- SYN 1 *a small planet and the farthest known planet from the sun; has the most elliptical orbit of all the planets*
- SYN 2 *(Greek mythology) the god of the underworld in ancient mythology; brother of Zeus and husband of Persephone*
- SYN 3 *a cartoon character created by Walt Disney*

Candidate nouns for concept Pluto

- SYN 1 planet; sun; orbit; planets;
- SYN 2 Greek; god; underworld; mythology; brother; Zeus; husband; Persephone;
- SYN 3 cartoon; character; Walt; Disney;

Patterns applied on SYN 1

Numbers of returned results are in brackets

-"Pluto is a planet" (1550),"Pluto is planet" (145)
-"Pluto is a sun" (2),"Pluto is sun" (0)
-"Pluto is a orbit" (0),"Pluto is orbit" (1)
-"Pluto is a planets" (0),"Pluto is planets" (0)

According to normalized numbers in brackets, Pluto is considered to be a candidate from the first pattern. The second pattern will confirm this fact in this case. Now we can search for "Pluto planet", "Pluto God" and "Pluto cartoon" to get **refined results as in Figure 1**.

5. TESTS

The test set consisted of 50 of proper nouns from space, travel and zodiac area. 96% (i.e. 48 out of 50) proper nouns have their glosses in WordNet. After all the tests have been carried out, it was necessary to check the correspondence of the discovered hypernym with the real life concepts (i.e. existing objects).

We discovered that from the test set, 62% (31 words which contained 61 synonymic classes in total) were assigned with a hypernym correctly and they corresponded to real life objects. 9 terms and all their meanings were assigned wrongly. The remaining 16% contained a mistake in assigning some of the synonym classes. More detailed results are in [6].

The progress that has been made since the last paper [6] is employing the Yahoo API. Each hypernym discovery requires about 40 queries. It took Google API more than 90 seconds to reply to the example answer. With Yahoo it was about 35 seconds, which is a significant time reduction. The second drawback discussed in the previous paper was the constraints of Google phrase search. The proximity search [5] allows to loosen constrains and search for more general query while retaining the exact positions of the query.

6. CONCLUSION

The first test showed that after some improvements the precision of 62% could be further improved. Employing more thorough tests and verifications and exploiting advanced features of fulltext search engines API would rise the precision and allow discovering hypernyms not only for a set of words from WordNet. We have already tested a version using proximate search to discover the most common adverbs for the given concept hypernyms.

Figure 1. Context specification for the searched word.

7. ACKNOWLEDGMENTS

The author would like to thank to Vojtech Svatek for his comments and help. The research has been partially supported by the FRVS grant no. 501/G1.

8. REFERENCES

[1] Baker L.:Search Engine Users Prefer Two Word Phrases, Search Engine Journal http://www.searchenginejournal.com/index.php?p=238

[2] Cimiano, P. and Staab S.:Learning by Googling. SIGKDD Explor. Newsl. 6, 2 (Dec. 2004), 24-33.

[3] Fellbaum C.: WordNet, an electronic lexical database, MIT Press, 1998.

[4] Hearst M. A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In Proceedings of the Fourteenth International Conference on Computational Linguistics,1992

[5] Kostoff RN, Rigsby JT, and Barth RB. Adjacency and proximity searching in the Science Citation Index and Google. DTIC Technical Report Number ADA 442 888

[6] Nemrava J.: Using WordNet glosses to refine search engine queries, In Dateso 2006, April 2006

[7] Porter M.: Porter Stemmer Algorithm, [online], http://tartarus.org/~martin/PorterStemmer/

[8] Ratnaparkhi A.: Adwait Ratnaparkhi's Research Interests, [online], http://www.cis.upenn.edu/~adwait/statnlp.html.

Knowledge acquisition, organization and maintenance for heterogeneous information resources

Nguyen G.
Institute of Informatics, SAS
Dubravska cesta 9
845 07 Bratislava
Slovakia
giang.ui@savba.sk

Laclavik M.
Institute of Informatics, SAS
Dubravska cesta 9
845 07 Bratislava
Slovakia
laclavik.ui@savba.sk

Babik M.
Institute of Informatics, SAS
Dubravska cesta 9
845 07 Bratislava
Slovakia
babik@saske.sk

ABSTRACT

The framework supporting data and knowledge acquisition, organization and maintenance for heterogeneous information resources is presented in this paper. It contains of the corporate memory (CM) and number of tools that work all together. The CM holds and manages documents, data and knowledge processed and created by tools. Tools work with data types e.g. documents, relational database and semantic data, etc. Each tool, in other view, can work also as independent tool to solve one specific problem. The framework finds its use in the automatic processing of documents with the aim to enable easy searching and finding information from wide space such as Internet. This whole chain process is quite complex and complicated with many non-trivial problems. The context of the knowledge management is based on the domain ontology which is a base for semantic data and creates a common background for entire system development.

1. INTRODUCTION

Nowadays, the Internet is becoming an universal repository of human knowledge which has allowed unprecedented sharing of ideas and information. Finding useful information is frequently a tedious and difficult. The difficulty is not only to know how to extract information, but also in knowing how to use it to decide relevance. The data retrieval process (as our project) aims to retrieving all objects which satisfy predefined conditions. At the moment, the approach of our framework is successful applied to the Job Offer search as the first pilot application of the NAZOU project [4], which enables users to find what they need, easily and adaptable to their profiles, preferences and requirements [2, 3]. A network of information is also provided and its services that can be processed by machines, which is different from the state of art, where web contain is mainly only human readable.

2. KNOWLEDGE CORPORATE MEMORY

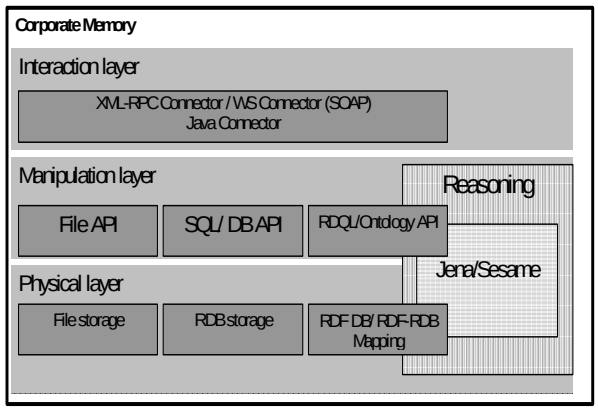


Figure 1: Cooperate Memory Architecture

Corporate Memory (CM) is accessible for other components using relevant client. The core of the CM is running as XML-RPC server and other components can call relevant client method via XML-RPC. CM is organized into three layers (Figure 1): physical layer (file system, database system, and ontological models), manipulation layer (access to the stored data and information) and interaction layer.

Ontology has become a very important aspect in many applications to provide a semantic framework to describe application domain. Ontology is a set of definitions of content-specific knowledge representation primitives (classes, relations, functions and constants). Ontology presents a shared understanding about a certain specific domain. There are also multiple inheritances, strong encapsulation, meta-data standards to train, discover and disambiguate meaning, and increased computing power. Although ontology enables processing knowledge and data, the most important role of ontology is in defining sharing meaning, emergence and discovery of gaps and for improving tacit knowledge transfer.

Semantic part of the CM is responsible for providing user interfaces for querying and manipulating the CM semantic content as well as providing the physical backend for persistent and transient storage of the semantics. The semantic model of the Web content is represented in the form of ontologies (OWL). The CM semantic part has two parts: the core interface (transparent access to the underlying knowledge repositories and reasoners) and the OntoClient inter-

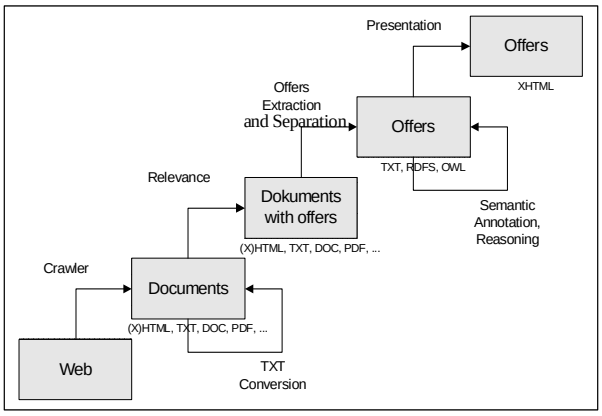


Figure 2: Chain of Tools

face (defines the possible interactions between the components of the system and the semantic part of the CM).

The relational DB management part of the CM is designed with virtualization concept, making actual DB system and DB connection object transparent to the client applications. The part of CM dedicated to file management provides a way for manipulating the file storage using unified application interface, making actual physical file storage transparent to the user or application. In the current implementation, CM's file storage is realized as a directory subtree of a file system directory tree. File management part of CM consist of core operations implementation and the client toolkit. Client toolkit can be configured to access the CM's file storage through local Java API with XML-RPC call or through Web Service interface, which is realized by OGSA-DAI framework.

3. DATA ACQUISITION, ORGANIZATION AND MAINTENANCE

Ontology based knowledge management includes activities like knowledge acquisition, creation, accumulation, sharing, reuse and capitalization. Knowledge items are abstracted to a characterization by metadata descriptions, which are used for further processing [1]. As it is described previously, here is a set of tools (Figure 2) that work with CM and each with other:

RIDAR (Relevant Internet Data Resource Identification): exploits the potential of existing search engines to identify relevant information resources on the Internet based on users-supplied search terms or more complicated search expressions. Details about identified resources (URL, title, etc.) are stored into databases.

WebCrawler traverses identified resources by RIDAR and downloads pages. These pages are then analyzed by ERID (Estimate Relevance of Internet Documents) tools, which estimate the page's relevance. The relevance estimation tires to decrease amount of downloaded documents by eliminating the pages with uninteresting content

DocConverter and DocIndexing transform documents from one to another format. At the moment, it transforms HTML to TXT documents for the need of other tools. The tool is accessible through WSRF standard compliant Web Service (WS) interface using OGSA-DAI framework. WS interface facilitates integration of the tool in distributed, heterogeneous environment.

ExPoS (Offer Extraction) processes downloaded and converted documents with offers and removes irrelevant information such as advertisements, etc. using several noise analysis methods. OSID (Offer Separation) separates blocks of offers from documents that contain more offers according to structure and offer identification indications. These two tools closely work together, they both deal with text processing and text analysis problem that have many non-trivial features. Their output is very important and userful for knowledge acquisition and organization.

Ontea (Ontology Based Text Annotation) annotates text version of offers by ontology individuals via regular expressions as relevant semantic properties of the offer then creates ontology form of offers according to predefined ontology. This can help e.g. in categorization, common visualization of documents, searching and knowledge inference or reasoning. While most of annotation solutions try to find and create an object in text or to provide semantic tags for a reader, Ontea tries to detect ontology elements within the existing domain ontology model. In this stage, experiments show the archived success of the tool around 80%.

4. CONCLUSIONS

In this paper, the chain of tools and functionalities of the CM of the framework is presented. Tools are almost independent but are integrated all together around the CM to achieve common aim as the whole. The approach is widely used in EU Research and Development and national projects for automatic data processing for knowledge management. At the moment, the work of our team is concentrated on enrichment and improvement of functionalities of tools and CM as well as the cooperation among them.

5. ACKNOWLEDGMENTS

This work is supported by NAZOU SPVV 1025/2004, EU RTD IST K-Wf Grid FP6-511385, APVT-51-024604 and VEGA No. 2/6103/6.

6. ADDITIONAL AUTHORS

Additional authors: Ciglan M. and Gatial E. and Balogh Z. and Oravec V. and Hluchy L.

7. REFERENCES

[1] R. Beaza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, Addison Wesley Longman, 1999.

[2] M. Bielikova and J. Kuruc and V. Marko. Entry into virtual university space through web-based e-application. *Elfa*, pages 403–410, 2004.

[3] R. Lencses. Indexing for the information retrieval system supported with relational database. In *Sofsem?05 Communications*, 2005.

[4] P. NAZOU Team. Nazou project website. In *http://nazou.fiit.stuba.sk/*, 2006.

Toward a Knowledge Base for Answering Causal Questions

Sodel Vazquez-Reyes
School of Informatics
The University of Manchester
P.O. Box 88, Manchester, M60 1QD, UK
+44 (0)161 30-62076
s.vazquez-reyes@postgrad.manchester.ac.uk

William J. Black
School of Informatics
The University of Manchester
P.O. Box 88, Manchester, M60 1QD, UK
+44 (0)161 30-63096
william.black@manchester.ac.uk

ABSTRACT
The use of lexicons has become common practice for most Natural Language Processing. Designing applications that interact with a pre-existing knowledge base –ontology, lexicon or both; could portend shorter development cycles and more scalable products. We propose to reuse the SUMO/WordNet mapping as a knowledge base for answering causal questions. Our analysis shows that this approach could lead to the retrieval of more accurate answers**.**

Categories and Subject Descriptors
I.2.7 [**Artificial Intelligence**]: Natural Language Processing; I.2.7 [**Knowledge Representation Formalisms and Methods**]: Representations (procedural and rule-based); I.7.1 [**Document and Text Processing**]

General Terms
Algorithms, Measurement, Experimentation

Keywords
Question Answering, Natural Language Processing, Knowledge Base, Causal Questions

1. INTRODUCTION
Open Domain Question Answering (ODQA) systems involve the extraction of answers –a phrase or a sentence, to a question rather than retrieval of relevant documents. ODQA research have been largely driven by the TREC QA track[1]; the type of questions to answer have been moving away from fact-based questions to more complex questions, which cannot be answered by simple name-entities. Research in ODQA has been mainly focused on responding factual questions, definition questions and list questions [6]. However, systems working with complex questions such as causal, procedural, comparative or evaluative are still under research.

[1] http://trec.nist.gov/data/qa.html

2. CAUSAL QUESTIONS
If we are working with causal questions, it is necessary to show why they are complex and how we could get answers to them; that is, our cause/effect approach. The majority of causal questions are the form 'Why *P*?', where *P* is an observation or fact to answer (which we have identified as an effect). If a 'why' question is an effect, then we are searching explanations for it (which we have identified as causes). For that reason, we have called to the cause and its effect causal relation [5].

Inappropriate answers to a question are mainly due to misunderstanding the questions themselves [2]. For example, the answer to the following question varies depending on how the question is understood.

Why did David eat dinner at the Mexican Restaurant?

If we understand that question concerns David's motivation for eating, we could reply that he ate there 'because he was hungry'. If we understand that the question relates to why David chose that particular restaurant, we could answer that 'He had heard that this is a good restaurant and he wanted to try it'. If we understand that the question is about David's going to a restaurant instead of eating at home, we could answer that 'David's wife is out of town and he can't cook'. We can observe that a 'why' question (effect) has infinite number of different answers (causes); we can find different causal relations.

Each answer contains an explanation of a cause for the question. These answers may be subjectively true, and each answer has boundary conditions for an answer. The accuracy of the answer is in the mind of the perceiver.

The original question terms can be used as the basis to locate potential answer candidates in the document collection; however, one major problem of doing this is that the question terms do not have sufficient coverage to locate most answer candidates. This process requires extra knowledge. Thus, we need to create a knowledge base that leads us to the retrieval of more accurate answers. Therefore, we focus on the kind of knowledge that could contribute to answer causal questions, and how the knowledge should be represented and used. The answers for causal questions involve judgments or evaluations; analyzing information, giving opinions or justifications, making predictions, interpreting situations or making generalizations. Furthermore, these questions require longer responses and can seldom be answered in one or two words.

3. KNOWLEDGE BASE DESCRIPTION

We have an ODQA architecture to answer causal questions, comprising three main components: *question analysis*, *document retrieval*, and *answer candidate extraction* [5]. This architecture performs an automatic pipeline lexico-semantic analysis using NLP techniques[1], such as: tokenization, POS tagging, sentence splitter, multi-word term extraction (statistical), ontology lookup (of single and multi-word names and terms) and BSEE compiler (context sensitive rule-based analysis to build representations of basic semantic elements of interest, events and relations). To be precise, a version of CAFETIERE designed for Question Answering which interacts with a knowledge base that is a phrasal lookup facility. This facility reads a prototype synonymy index to an ontology from memory instead of a database. The index is currently maintained as a plain text file, although our goal is to maintain it using Protégé or another equivalent editing tool, and so to provide a CAFETIERE's plug-in that will update the index. The index can map different synonyms or abbreviations to a standard class or instance name, and also read and store features and values either through ontology lookup or BSEE compiler.

The resources that we are evaluating are the Suggested Upper Merged Ontology (SUMO) a domain-independent upper level ontology which focuses on promoting data interoperability, information search and retrieval, automated inference and natural language processing [4]. WordNet lexicon is a valuable resource for automated processing of natural language [3].

A big advantage of SUMO is that it has already been mapped to the entire WordNet lexicon [4]. This means each synset is tagged with the corresponding SUMO concept. SUMO and WordNet define conceptualizations. Through WordNet, we can map conceptualizations into a natural language multiword term of one or more words, and with SUMO we can organize them into a logical structure. In other words, SUMO/WordNet mapping allows us map natural language words into SUMO concepts, using WordNet synsets as an intermediate layer. Our goal is to use SUMO/WordNet mapping as a knowledge base for our ODQA architecture as previously mentioned.

SUMO/WordNet mapping uses three relations: equivalent, subsuming, and instance [4]. Consider the following example for clarification purposes:

07544210 13 n 01 buffalo_wing 0 001 @ 07454864 n 0000 | crisp spicy chicken wings &%Food+

The '&%' prefix indicates that the term 'food' is taken from the SUMO ontology, and the suffix '+' indicates that the concept is a hypernym of the associated synset. When synset is equivalent with the SUMO concept, the suffix is '=', which indicates that the mapping relation is synonymy. The instance relation indicates that the thing denoted by the WordNet synset is a member of the class denoted by the SUMO concept; its suffix is '@'.

We have developed a text mining application that uses the SUMO/WordNet mappings files, SUMO KIF files and WordNet lexicon to populate the knowledge base used into our ODQA system, following CAFETIERE index format to take advantage of this knowledge base. The proposed structure is:

TERM<id><relation><SUMO class><WordNet synset><WordNet gloss><WordNet hypernym hierarchy><SUMO superclass hierarchy><CAFETIERE features>

A separate line of data is given to each term in the knowledge base. That term could have different meanings and different syntactic categories: noun, verb, adjective, or adverb, which is also why a separate line is given to each one. CAFETIERE features are semantic relations –domain_usage, verb_group, entailment, cause, similar_to, and attribute. We use three special characters: '>' like separator between elements, ':' like separator into the hierarchies, and '_' represents a white space. Let us show an example using the term *"buffalo wings"* in order to clarify the proposed structure.

buffalo_wing>07544210buffalo_wing>relation=subsuming>sc=Food>wns=buffalo_wing>wng=crisp_spicy_chicken_wings>wnh=dish:nutriment:food:substance:entity>ssc=SelfConnectedObject:Object:Physical:Entity>wnu=null>wna=null

We consider such data an essential knowledge base for performing text analysis with ontological semantic interpretation in answering causal questions. However, strictly speaking, a knowledge source is only ever partial in practice, covering just a subset of domain.

4. CONCLUSION AND FUTURE WORK

We have described a knowledge base as a general purpose ontology, in order to make a stronger ontological semantic interpretation in answering causal questions. The initial cycle of knowledge base preparation has already been implemented. Currently, we are evaluating its contribution into the automatic pipeline lexico-semantic analysis for questions and documents, in order to detect potential further improvements.

5. ACKNOWLEDGES

Thanks to Articulate Software and Teknowledge for providing source code of Sigma knowledge engineering system.

6. REFERENCES

[1] Black, W. J., McNaught, J., Vasilakopoulos, A., Zervanou, K., Theodoulidis B., and Rinaldi F. *CAFETIERE: Conceptual Annotations for Facts, Events, Terms, Individual Entities, and Relations*. Technical Report of PARMENIDES Project, The University of Manchester, 2003.

[2] Galambos, J. A., and Black J. B. Using Knowledge of Activities to Understand and Answer Questions. In A. C. Graesser & J. B. Black (Eds.), *The psychology of questions*. Hillsdale, NJ: Erlbaum, 1985.

[3] Miller, G. A. *WordNet: A Lexical Database*. Communication of the ACM 38(11):39-41, November 1995.

[4] Niles, I., and Pease, A. Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In *Proceedings of the International Conference on Information and Knowledge Engineering 2003*.

[5] Vazquez-Reyes, S., and Black, W. Building a Framework for Answering WHY Questions. In *Proceedings of the 9th Annual Colloquium on Computational Linguistics,CLUK'06.*

[6] Voorhees, E. M. Overview of the TREC 2004 Question Answering Track. In E. M. Voorhees and D. K. Harman, editors, *Proceedings of the Twelfth Text Retrieval Conference.* NIST Special Publications 500-261.

Structural Techniques for Alignment of Structurally Dissymmetric Taxonomies

Chantal Reynaud, Brigitte Safar, Hassen Kefi
LRI-PCRI Batiment 490 Université Paris-Sud
91405 Orsay Cedex France
firstname.lastname@lri.fr

ABSTRACT
This paper deals with taxonomy alignment and presents the structural techniques of an alignment method suitable with a dissymmetry in the structure of the mapped taxonomies. The aim is to allow a uniform access to documents belonging to a same application domain, assuming retrieval of documents is supported by taxonomies. We applied our method to various taxonomies using our prototype TaxoMap.

Categories and Subject Descriptors
H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*

General Terms
Design, Algorithms, Experimentation

Keywords
Taxonomy, Alignment, Mapping, Unified Access

1. INTRODUCTION
Our work focuses on taxonomy alignment techniques. Indeed, we assume that description of content of most information systems is often based on very simple ontologies reduced for the present to classification structures, i.e. taxonomies. Moreover, we suppose that the structures of the taxonomies that we align are heterogeneous and dissymmetric, one taxonomy being deep whereas the other one is flat. Such a situation can be encountered for example when we try to access to additional resources with very simple classification structures describing the domain concepts from a Web portal having its own query interface based on a hierarchically well-structured taxonomy. In this context, the approaches that rely on OWL data representations, exploiting all the ontology language features, don't apply [3]. Similarity of two entities can't be identified based on their properties or on the status of their parents and siblings because this information is not available. To find mapping candidates between structurally dissymetric taxonomies, we can only use the following available data: labels of concepts in both taxonomies, the structure of the deeper taxonomy and external resources such as WordNet.
This paper describes two structural techniques designed to make best use of the characteristics of the taxonomies: very specialized taxonomies with only subclass links, concepts with labels which are expressions composed of a lot of words, words common to a lot of labels. These techniques have been evaluated on real-world taxonomies and on test ones extracted from a repository about ontology matching [5]. Experiments showed that the proposed techniques give very relevant mappings when the aligned taxonomies have the same characteristics as those having motivated our approach.

2. THE ALIGNMENT PROCESS
For us, a taxonomy is a pair (C, H_C) consisting of a set of concepts C arranged in a subsumption hierarchy H_C. A concept is only defined by two elements: a label and subclass relationships. The label is a name (a string) that describes entities in natural language and that can be an expression composed of several words. Subclass relationships establish links with other concepts. It is the single semantic association used in the hierarchy.
Given two structurally dissymmetric taxonomies, our objective is to map the concepts of the less structured one, the source taxonomy T_{Source}, with concepts of the more structured one, the target taxonomy T_{Target}.The alignment process is oriented from T_{Source} to T_{Target}. It aims at finding one-to-one mappings which are relations of two kinds: equivalence ($isEq$) and subclass (isA). So, for each concept c_S in T_{Source}, we try to find a corresponding concept c_T in T_{Target} linked to c_S with an equivalence or a subclass relation.

3. THE ALIGNMENT TECHNIQUES
3.1 General view
Alignment is based on Lin's similarity measure [1], computed between each concept c_S in T_{Source} and all the concepts of T_{Target}. This measure compares strings and has been adapted to take into account the importance of the words inside the expressions. Various techniques are applied in sequence to make the overall alignment process the most efficient as possible. For each technique, the objective is to select the best concept in T_{Target} among a lot of mapping candidates (with a similarity measure not null). This best concept is not necessarily the concept with the highest similarity measure. We classify the found mappings into two groups according to their relevance: probable mappings and

potential mappings to be confirmed.

Algorithm 1: Alignment process
$TaxoMap(T_{Source}, T_{Target})$
1. **For each** $c_S \in T_{Source}$ **do**
2. **For each** $c_T \in T_{Target}$ **do** $Sim_{LinLike}(c_S, c_T)$
3. $MC \leftarrow$ MappingCandidates(c_S)
4. **If** ProbableMapping(c_S, MC) **then** stop
5. **Else** PotentialMapping(c_S, MC)

Terminological techniques are executed first. In default of place, they will not be detailed here. Being based on the richness of the labels of the concepts, they provide the most probable mappings (cf. Alg.1). However a lot of mappings are not found. So we propose to complete these first techniques with two structural ones suited to our work context, deriving interesting but less sure (potential) mappings. A user evaluation of these new mappings is necessary whereas the evaluation of mappings of the first group is not or can be done very quickly.

3.2 Exploiting structural features

The two structural techniques that we propose are complementary: the first one is operative when labels are composed of many words, the second one maps concepts with short labels (one or two words).

3.2.1 *Exploiting the structure of* T_{Target}

This first technique, STR_T, works on MC, the set of mapping candidates of a concept c_S. MC includes concepts with a high similarity value with c_S (only the three most similar concepts b_1, b_2, b_3 are retained) and Inc, the set of concepts of T_{Target} with a label included in the label of c_S. The idea is to exploit the location of the mapping candidates in T_{Target}. If a great number of elements in MC has a common ancestor which is deep enough in T_{Target}, that means that those elements are close and share a common context, and we assume that c_S is meaningful according to that context too. That way we avoid mappings with isolated candidates meaningful in another context, whose similarity measure is a little higher. The concept retained for the mapping with c_S belongs to the common context and has the highest similarity value. It is a possible father or a brother of c_S depending on whether it belongs to Inc or not.

3.2.2 *Exploiting the structure of WordNet*

The second technique, STR_W, exploits the hyperonymy /hyponymy WordNet structure in order to map concepts which are semantically similar without being synonyms and whose labels are syntactically different. This technique can, for example, map cantaloupe with watermelon which are not synonyms but two specializations of melon in WordNet.
The use of WordNet is as follows. An expert identifies the application root node, noted $root_A$, that is the most specialized concept in WordNet which generalizes all the concepts of the concerned application domain. Then we search for the hypernyms in WordNet of each term of T_{Source} not yet mapped and of each term of T_{Target} (according to all their senses) until $root_A$ or the top of WordNet is reached. Only the paths from the invoked terms to $root_A$ will be selected because they represent the only accurate senses for the application. That way, a sub-tree, called T_W, is obtained. It is composed of all the terms and the relations of the retained paths. For each concept c_S, we select in T_W the most similar concept belonging to T_{Target} using Wu and Palmer's measure [4]. This selection is very efficient because it doesn't require much similarity computation [2].

4. EXPERIMENTS AND DISCUSSION

Two kinds of experiments have been performed. First, experiments have been made in the setting of the e.dot project[1], on two real-world taxonomies in the field of predictive microbiology. Second, we applied our techniques on test taxonomies [5]. The latter are not structurally dissymmetric and cover a large domain. The application conditions of the techniques are not achieved but our objective is to make these tests in order to sketch some ideas to do improvements and to widen the scope of our approach. These experiments have shown where our specific strengths and weaknesses are. Whatever taxonomy we aligned, our approach was able to retrieve almost all the expected equivalence mappings. Furthermore, its strong point is to propose in addition a lot of other mappings (subclass mappings). Some mappings have a high precision and are then sure (probable mappings generated by the terminological techniques). Other ones (potential mappings generated by the structural techniques) are less sure (low precision) and have to be validated. This confirms the order in the application of our techniques. Concerning the structural techniques, STR_T proved to be very useful and leads to relevant mappings when concepts have labels composed of a lot of words and when some words are common to many labels. On the opposite, STR_W is all the more appropriate since the application domain is small. The real-world taxonomies which have motivated our approach gather all these characteristics, unlike the others. Then better results are obtained.

5. CONCLUSION

We described two structural techniques to align structurally asymmetric taxonomies. These techniques are original because different from a search of structural similarity in models. They are executed to suggest additional mappings. These mappings are not sure but they can be a good complement, if human involvement is possible, as experiments showed it. We will continue this work by adapting and extending our techniques according to the experiment results. Our first objective is to be able to align taxonomies relative to larger application domains.

6. REFERENCES

[1] D. Lin. An Information-Theoretic Definition of Similarity, In *Proc. of the Int. Conf. on Machine Learning (ICML)*, pp. 296-304, 1998.

[2] C. Reynaud and B. Safar. Structural techniques for alignment of taxonomies: experiments and evaluation, Technical Report 1453, LRI, Univ. of Paris-Sud, June 2006.

[3] P. Shvaiko, J. Euzenat. A Survey of Schema-based Matching Approaches, In *J. on Data Semantics*, 2005.

[4] Z. Wu and M. Palmer. Verb Semantics and Lexical Selection, In *Proc. of 32nd Meeting of the Ass. for Computational Linguistics*, 1994.

[5] http://www.ontologymatching/evaluation.html

[1]E.dot is a research project funded by national network on software technology (RNTL), 2003-2005.

Exploring Pathways Across Stories

Zdenek Zdrahal
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA, UK
Z.Zdrahal@open.ac.uk

Paul Mulholland
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA, UK
P.Mulholland@open.ac.uk

Trevor Collins
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA, UK
T.D.Collins@open.ac.uk

ABSTRACT

This paper describes a method for supporting the exploration of a collection of documents organized as a hypertext by investigating relations between documents along user-specified paths. The approach is demonstrated on a corpus of stories about the WW2 activities of the British Government Code and Cypher School at Bletchley Park. Each story is described by one or more events and annotated in terms of domain ontologioes. A pathway in the document space is a sequence of events in which adjacent events share common binding concepts. The criteria for selecting the pathway include a measure of the adherence to the user-specified part document space and the mutual information between adjacent documents calculated from their annotations.

Categories and Subject Descriptors

H.3 [**Information storage and retrieval**]: Content Analysis and Indexing, Web-based services, Collection

1. INTRODUCTION

This paper describes methods for exploration of a collection of stories. The term "story" refers to a semantically self-contained block of text, possibly with associate pictures or multimedia. There are two main reasons for concentrating on documents in the form of stories: knowledge is often represented in stories [4], and there is the natural way of breaking the document into smaller units - events. We assume that the documents in the collection are semantically interrelated i.e. they share common key concepts and refer to related events. The whole collection can be regarded as a form of hypertext. However, unlike standard hypertext, the links between *lexias* are not explicitly predefined, but are dynamically constructed in accordance with the user's needs from the annotations of documents. In addition to the knowledge extracted from individual documents, further meaning can be inferred from organizing and presenting documents in different structures and in this way facilitate the discovery of the knowledge hidden in the collection.

Posters and Demos of the 15th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2006 Podebrady, Czech Republic, 2nd-6th October, 2006

2. CASE STUDY: BLETCHLEY PARK

2.1 The Bletchley Park Story

During World War Two, Bletchley Park was the location of the Headquarter of the British Government Code and Cypher School and hosted a number of distinguished scientists who worked on breaking enemy codes. In the early 1990s the place was converted into a museum and the tour guides started collecting documents about the Bletchley Park history. At present, the collections consist of thousands of unique documents about code breaking, early computing, life and work of prominent scientists and ordinary staff in Bletchley, the impact of the Bletchley Park effort on the course of the war and other similar topics.

2.2 Bletchley Park Text

For content exploration by analyzing pathways about 400 most interesting stories were selected by the tour guides. The application developed for the museum is called *Bletchley Park Text*. Stories were annotated in terms of domain ontologies, with the CIDOC Conceptual Reference Model (CRM) [1] as the upper ontology . The next level ontologies include the *bletchley-park-ontology* which specialize the CRM concepts for the Bletchley Park Text and the *narrative-ontology* which defines simple narrative concepts that are used to specify associations between concept and presentation levels. The dynamic links between annotated documents (stories or events) are defined in terms of binding concepts which are instances of classes *actor*, *place* stored in knowledge bases. Pathway is a sequence of stories in which two adjacent stories share a binding concept. Slot types are ignored for defining paths, but are used for interpreting paths.

3. FORMAL REPRESENTATION

The document space can be represented as a hypergraph $H =< C, E >$, where $C = \{c_1, c_2, \ldots, c_N\}$ is a set of nodes corresponding to concepts and $E = \{e_1, e_2, \ldots e_M\}$ is a set of edges corresponding to events (stories, documents). The document hypergraph is constructed as follows: (1) Annotated events in all documents specify the set of edges $E = \{e_i\}$. Edges are n-tuples of concepts with associated event names for easy identification. (2) The set of nodes $C = \{c_j\}$ is defined as a union of all edges $C = \cup e_i$. The slot names of events are not used.

4. EXPLORATION OF PATHWAYS

There are often many pathways between selected concepts and therefore some strategy for their ranking is needed. De-

pending on the ranking criterion, pathways may highlight different properties of the document space and play different roles in content exploration.

4.1 Focusing document space

The document space typically describes many interrelated themes. Each theme is associated with a cluster of documents that share many common concepts. The users are often interested only in a specific theme, however they would like to explore this theme in detail. They can chose their theme of interest by marking a few *seed concepts* which restrict the document space and focus the exploration. Based on a selected set of seed concepts the document space is reconstructed as follows: Let $S = \{c_1, \ldots, c_s\}$ be a set of seed concepts selected from the set of nodes C. Let us denote as $E_s = \{e_1, \ldots, e_s\}$ the set of all hypergraph edges that contain at least one concept of S and C_s the set constructed as already described in Section 3. Then hypergraph $H_s =< C_s, E_s >$ is a partial hypergraph of H and the corresponding subspace of the original document space is called *focused document space*. If all seed concepts belong to the same theme, then the focused document space contains only concepts of this theme and paths might be constructed only from concepts, events and stories of this theme. However, as the clusters (themes) are overlapping seed concepts may select multiple themes.

4.2 Exploring focused document space

Paths in the focused document space are constructed by the same algorithm as in the original space. In the focused document space, the shortest path is longer or of the same length as the original one. We have implemented two algorithms for guiding exploration by constructing shortest paths in the focused document space: (1) Only nodes and edges of the focused document space are used. (2) If possible, nodes and edges of the focused document space are used. If such a pathway cannot be constructed, concepts from the original document space might be included and the step is penalized.

5. MUTUAL INFORMATION IN DOCUMENT SPACE

Each exploratory step along the path in the document space is associated with acquiring new information. In each step, the information shared by two events can be measured by mutual information, defined as $I(c_i : c_j) = \log_2 \frac{P(c_i,c_j)}{P(c_i)\cdot P(c_j)}$, where $P(c_i)$ and $P(c_j)$ are probabilities of c_i and c_j respectively, and $P(c_i, c_j)$ is the joint probability of c_i and c_j. Probabilities are calculated as relative frequencies, i.e. $P(c_i) = \frac{|E(c_i)|}{|E|}$.

5.1 Information based criterion in a focused document space

The information criterion can quantitatively evaluate paths and can be applied both to exploration of the complete document space and to the focused document space. Focusing document space affects the values of mutual information along the path. The number of events is not reduced evenly across the document space. In particular, the number of events in sets $E(c_i)$, $E(c_j)$ and $E(c_i \& c_j)$ does not change, but focusing removes unrelated concepts and therefore reduces the total number of events from E to E_F, see figure

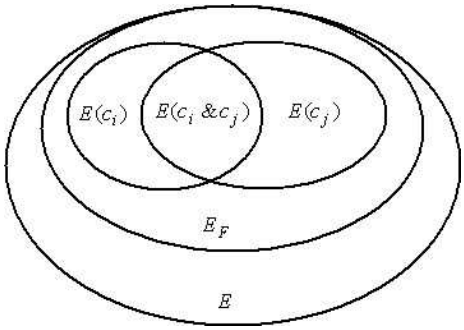


Figure 1: Reducing document space by focusing

1. If $P_F(.)$ is probability, defined as a relative frequency and $I_F(c_i : c_j)$ is mutual information in the focused space, we can easily show that $I(c_i : c_j) = I_F(c_i : c_j) + \log_2 \frac{|E|}{|E_F|}$. The difference $\log_2 \frac{|E|}{|E_F|}$ is the information gained by focusing the document space, e.g. by stating that stories from $E - E_F$ have been excluded.

6. CONCLUDING REMARKS

Content exploration is an important task of knowledge management. Recently, a special issue of the Communications of the ACM was dedicated to this theme, e.g. see [3] and [2]. Using pathways for navigation and exploration of a document set of has a long tradition and goes back to the memex ideas of Vannevar Bush. In this paper we have introduced two methods supporting pathway analysis: focusing of the document space and mutual information as a measure for the relationship between two adjacent documents.

The complete Bletchley Park Text application includes a number of different techniques including the pathway exploration. The application is routinely used in Bletchley Park since April 2005. During the museum tour, the visitors may express their interest by sending a text message from their mobile phones to a dedicated phone number. The text message may contain a few concepts they would like to study. After returning home, they can login into the Bletchley Park Text web site, use their mobile phone number as a password and start their own customized post-visit content exploration. The complete application will be demonstrated at the EKAW conference.

7. REFERENCES

[1] *CIDOC Conceptual Reference Model*. Proposal for ISO 21127: A Reference Ontology for the Interchange of Cultural Heritage Information, http://cidoc.ics.forth.gr/, 2004.

[2] E. Fox, F. D. Neves, Y. Xiaoyan, R. Shen, S. Kim, and W. Fan. Exploring the computting literature with visualisation and stepping stones and pathways. *Communications of the ACM*, 49(4):52–58, April 2006.

[3] M. Hearst. Clustering versus faceted categories for information exploration. *Communications of the ACM*, 49(4):59–61, April 2006.

[4] R. C. Schank. *Tell me a story*. Northwestern University Press, Illinois, 2000.

Author Index

Title: EKAW 2006

Type of publication: Book of abstracts
Submitted: authors, co-authors
Format: A4
Number of pages: 50
Year of issue: 2006
Edition: first

Published by: Zeithamlová Milena, Ing. - Agentura Action M
Vršovická 68
101 00 Praha 10
actionm@action-m.com
http://www.action-m.com

Printed by: Reprostředisko UK MFF
Sokolovská 83
186 75 Praha 8

No editorial and stylistic revision.

Not for sale.

ISBN 80-86742-15-6

*The **15th International Conference on Knowledge Engineering and Knowledge Management** was concerned with all aspects of eliciting, acquiring, modelling, managing and exploiting knowledge, and the role of these aspects in the construction of knowledge-intensive systems and services. The conference was held on 2nd – 6th October 2006 in Poděbrady, Czech Republic. This volume contains 21 extended abstracts for posters and demos.*

ISBN 80-86742-15-6