SEMAPRO 2011

The Fifth International Conference on Advances in Semantic Processing

ISBN: 978-1-61208-175-5

November 20-25, 2011

Lisbon, Portugal

SEMAPRO 2011 Editors

Pascal Lorenz, University of Haute Alsace, France

Eckhard Ammann, Reutlingen University, Germany

SEMAPRO 2011

Foreword

The Fifth International Conference on Advances in Semantic Processing [SEMAPRO 2011], held between November 20 and 25, 2011 in Lisbon, Portugal, constituted the stage for the state-of-the-art on the most recent advances in ontology, web services, semantic social media, semantic web, deep semantic web, semantic networking and semantic reasoning.

Semantic processing considers contextual dependencies and adds to the individually acquired knowledge emergent properties and understanding. Hardware and software support and platforms were developed for semantically enhanced information retrieval and interpretation. Searching for video, voice and speech [VVS] raises additional problems to specialized engines with respect to text search. Contextual searching and special patterns-based techniques are current solutions.

We take here the opportunity to warmly thank all the members of the SEMAPRO 2011 Technical Program Committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to SEMAPRO 2011. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the SEMAPRO 2011 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that SEMAPRO 2011 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the area of semantic processing.

We are convinced that the participants found the event useful and communications very open. We also hope the attendees enjoyed the historic charm of Lisbon, Portugal.

SEMAPRO 2011 Chairs:

Bich-Lien Doan
Nima Dokoohaki
Peter Haase
Wladyslaw Homenda
Thorsten Liebig
René Witte

SEMAPRO 2011

Committee

SEMAPRO Advisory Chairs

René Witte, Concordia University - Montréal, Canada
Wladyslaw Homenda, Warsaw University of Technology, Poland
Bich-Lien Doan, SUPELEC, France

SEMAPRO 2011 Industry Liaison Chairs

Peter Haase, Fluid Operations, Germany
Thorsten Liebig, derivo GmbH - Ulm, Germany

SEMAPRO 2011 Research Chair

Nima Dokoohaki, Royal Institute of Technology (KTH)-Kista, Sweden

SEMAPRO 2011 Technical Program Committee

Nasser Alalwan, King Saud University - Riyadh, Saudi Arabia
José F. Aldana Montes, University of Málaga, Spain
Sofia J. Athenikos, Drexel University, Philadelphia, USA
Sören Auer, AKSW-Universität Leipzig, Germany
Isabel Azevedo, ISEP-IPP, Portugal
Ebrahim Bagheri, Athabasca University and National Research Council Canada
Helmi Ben Hmida, FH MAINZ, Germany
Janez Brank, Jozef Stefan Institute, Ljubljana, Slovenia
Christopher Brewster, Aston University - Birmingham, UK
Ozgu Can, Ege University, Turkey
Tru Hoang Cao, Vietnam National University - HCM & Ho Chi Minh City University of Technology, Vietnam
Sana Châabane, ISG - Sousse, Tunisia
Sam Chapman, K-Now Limited, UK
Paolo Ciancarini, Università di Bologna, Italy
Smitashree Choudhury, DERI/National University of Ireland - Galway, Ireland
Jan Dedek, Charles University in Prague, Czech Republic
Alexiei Dingli, The University of Malta, Malta
Bich-Lien Doan, SUPELEC, France
Milan Dojčinovski, Czech Technical University in Prague, Czech Republic
Nima Dokoohaki, Royal Institute of Technology (KTH) - Stockholm, Sweden
Raimund Ege, Northern Illinois University, USA
Anna Fensel, FTW Forschungszentrum Telekommunikation Wien GmbH, Austria
Enrico Francesconi, ITTIG - CNR - Florence, Italy
Alexandra Galatescu, National Institute for R&D in Informatics - Bucharest, Romania
Stefania Galizia, INNOVA S.p.A., Italy
Raúl García Castro, Universidad Politécnica de Madrid, Spain
Rosa M. Gil Iranzo, Universitat de LLeida, Spain
Sotirios K. Goudos, Aristotle University of Thessaloniki, Greece
Gregor Grambow, Aalen University, Germany

Fabio Grandi, University of Bologna, Italy
Daniela Grigori, University of Versailles, France
Alessio Gugliotta, Innova SpA, Italy
Peter Haase, Fluid Operations, Germany
Ivan Habernal, University of West Bohemia - Plzen, Czech Republic
Ralf Heese, Freie Universität Berlin, Germany
Christian F. Hempelmann, RiverGlass, Inc., USA / Purdue University, USA
Cory Andrew Henson, Wright State University, USA
Wladyslaw Homenda, Warsaw University of Technology, Poland
Carolina Howard Felicissimo, BRGC - Schlumberger Brazil
Prasad Jayaweera, University of Sri Jayewardenepura, Sri Lanka
Wassim Jaziri, ISIM Sfax, Tunisia
Ivan Jelinek, Czech Technical University - Prague, Czech Republic
Katia Kermanidis, Ionian University - Corfu, Greece
Jacek Kopecky, The Open University, UK
Jaroslav Kuchar, Czech Technical University in Prague, Czech Republic
Vitaveska Lanfranchi, University of Sheffield, UK
Kyu-Chul Lee, Chungnam National University - Daejeon, South Korea
Thorsten Liebig, derivo GmbH - Ulm, Germany
Dong Liu, The Open University, UK
Sandra Lovrencic, University of Zagreb - Varaždin, Croatia
Maria Maleshkova, The Open University, UK
Paola Mello, DEIS - University of Bologna, Italy
Elisabeth Métais, Cedric-CNAM, France
Vasileios Mezaris, Centre for Research and Technology Hellas (ITI-CERTH), Thermi-Thessaloniki, Greece
Malgorzata Mochól, T-Systems Multimedia Solutions GmbH - Berlin, Germany
Shahab Mokarizadeh , Royal Institute of Technology (KTH) - Stockholm, Sweden
Ekawit Nantajeewarawat, Sirindhorn International Institute of Technology / Thammasat University, Thailand
Sasa Nesic, IDSIA/University of Lugano, Switzerland
Andriy Nikolov, The Open University, UK
Lyndon J. B. Nixon, STI International, Austria
Laura Papaelo, University of Genova, Italy
Carlos Pedrinaci, The Open University, UK
Andrea Perego, Università degli Studi dell'Insubria - Varese, Italy
Jaime Ramirez, Universidad Politécnica de Madrid, Spain
Isidro Ramos, Valencia Polytechnic University, Spain
Juergen Rilling, Concordia University - Montreal, Canada
Tarmo Robal, Tallinn University of Technology, Estonia
Sérgio Roberto da Silva, Universidade Estadual de Maringá, Brazil
Dilletta Romana Cacciagrano, University of Camerino, Italy
Thomas Roth-Berghofer, University of West London, UK
Michele Ruta, Politecnico di Bari, Italy
Melike Sah, University of Dublin, Ireland
Satya Sahoo, Wright State University, USA
Munehiko Sasajima, Osaka University, Japan
Minoru Sasaki, Ibaraki University, Japan
Najla Sassi, ISSAT Mahdia, Tunisia
Christoph Schmitz, 1&1 Internet AG - Karlsruhe, Germany
Kinga Schumacher, German Research Center for Artificial Intelligence (DFKI) - Berlin, Germany
Md. Sumon Shahriar, Tasmanian ICT Centre/CSIRO, Australia
William Song, Durham University, UK
Sofia Stamou, Ionian University, Greece
Mari Carmen Suárez-Figueroa, Universidad Politécnica de Madrid (UPM) , Spain

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Qualifying Audiovisual Searching Results with Ontologies and Semantic Algorithms

Luiz Rolim, Ismael Ávila, Alexandre Osorio
Service Technologies Department
CPqD R&D Foundation
Campinas, São Paulo, Brazil
[lrolim, avila_an, aosorio]@cpqd.com.br

Abstract— **Multimedia capabilities in end-user terminals, improvements on audiovisual (AV) encoding technologies and the ease of handling AV contents in the Internet have all contributed to the growing use of this media on the Web. Nowadays, searching for videos has become as common as searching for documents, news, web pages or other types of media, being the amount of non-relevant results returned as response to user's queries a common problem posed by the majority of searching engines. Among the myriad of approaches under consideration for qualifying the results of the queries, the usage of semantic technologies is one of the most attracting techniques. In this work, we present how an OWL ontology of subjects, or themes, can improve the efficiency of searching engines through the adoption of semantic algorithms operating over selected contents metadata descriptors based on DCMI and MPEG-7 standards. The main goal is to develop an algorithm that explores the semantic relationships of the supporting ontology and allow searching engines to return results that more appropriately match the actual interest of the end-users.**

Keywords-ontology; metadata; audiovisual; content searching; semantic algorithm.

I. INTRODUCTION

The advent of the Internet, of the high speed networks and multimedia capabilities on personal devices has lead to a widespread production of audiovisual (AV) contents by companies, institutions and end-users. In addition to text and static images, videos are employed pervasively as a way of documenting facts and situations of the everyday life, as well as a tool to transmit messages, express points of view or for artistic purposes. Today's Web provides various options for video search or video sharing, such as YouTube, Mubi and Vimeo. A common aspect encountered in these services is that the searching options can take advantage of the metadata description which generally accompanies the AV contents. Since the metadata contain specific information regarding the production, context, protagonists, themes and other aspects of content, specialized searching engines can provide advanced searching and presentation options when compared to traditional searching engines based on generic text comparisons. However, even on the AV specialized searching tools mentioned above, the adopted cataloguing and searching models are generally based on plain text descriptions and semantic-less keywords or categories, returning results that are non-qualified from a semantic perspective. As a consequence, in large repositories, many of the returned items may not meet the actual interests of the user.

Concurrently with the on-going efforts on improving the efficiency of existing searching engines, semantic-based technologies could play an important role in video browsing and cataloguing as described in [1] and [2], studies based on structured sets of metadata descriptors, such as MPEG-7 [3] and Dublin Core Metadata Initiative - DCMI [4], and on a supporting ontology. The work presented here-in adopts a similar approach that aims at developing algorithms capable of exploring the semantics of video content metadata. In this paper, we present a proposal for a video searching semantic algorithm and the structure of the supporting ontology.

Due to its inherent simplicity and its widespread use as a resource description scheme, we selected DCMI as the metadata standard for the overall description of the contents. For the description of specific AV elements, such as video-segments, the choice was for MPEG-7, which provides a comprehensive descriptor set to represent specific AV content structures. Together, the two descriptor sets form an application profile similar to the one described in [5]. Semantic capabilities are provided through a supporting ontology containing structured subject terms which are made available to cataloguing and searching tools.

This research is an activity of the Experimental TV project, a part of the GIGA R&D (Research & Development) program, consisting of a high speed optical network and associated services, currently being developed by the CPqD Foundation (www.cpqd.com.br), a Brazilian R&D Center. The research comprises the elaboration of the supporting OWL (Web Ontology Language) [6] ontology, the semantic cataloguing and searching tools, and a field experiment with community TVs and independent video producers [7]. The goal is to evaluate how semantic enabled searching engines can provide more qualified results to end users and make the searching process more effective. At the same time, it will be observed the influence of semantic enabled cataloguing and searching tools in promoting the sharing of video contents and the participation of end-users in an established video description collaboration process.

The remainder of the paper is organized as follows. Section II discusses related works on the area of semantic video searching. In Section III, the cataloguing process for AV contents is presented together with the structure of the supporting ontology. Section IV describes the semantic algorithm proposed, exemplified with a use case in Section V. The paper ends with conclusions and a discussion on further work.

 ISBN: 978-1-61208-175-5

II. RELATED WORK

To correctly situate the solution proposed herein in the area of semantically enabled search [14] (SES) it is first necessary to review some current approaches in SES. An SES solution can be aimed to solve different types of search, depending on how narrow is the initial target defined by the user, or even depending on whether there is a clear target at all. In the so-called "navigational searches", the users know precisely what content they are looking for, and the process of finding it is navigating (browsing) to that particular document. In the "exploratory searches", on the other hand, the users have no precise idea of what will be the outcome of the search, probably because they are not familiar with the topic being searched, and their interests can change as they are presented to new search results. In between these two extremes one can distinguish "research searches" [13], where the users have some topics in mind, but no particular document.

According to [16], one can consider exploratory searches as a specialization of information exploration, and interface features such as dynamic queries can help users to see the immediate result of their decisions. To evaluate such systems it is necessary, for instance, to compare the time spent in finding and selecting the information. The solution for exploratory video searching described in [14] combines results from a specific video index with complementary data from DBpedia, which is an initiative to semantically structure information from Wikipedia and dispose the results on the Web. In order to determine, for the query string, a list of related entities, a set of heuristics are applied to the entities in DBpedia. The objective is to determine the relevance of one property based on the frequency it occurs on instances of a category or type in DBpedia. The resources suggested to the user are the ones connected to the highest frequent properties and that are available in the video index. Another approach to the problem is to conceive search engines totally based on the Semantic Web, such as the one described in [15].

In this work, our expectation is to contribute to the audiovisual searching area with the conception of semantic algorithms supported by an OWL ontology containing the knowledge to be applied to the searching process. Overall, we expect to explore functionalities that provide benefits in all the search categories described above.

III. METADATA AND ONTOLOGY

The infrastructure for the semantic AV content searching engine consists of a database containing the metadata descriptors and the supporting ontology. The AV contents files may be stored in one or more repositories and the access to the content is ruled by property rights defined by the owner(s). Searching results comprise an URL providing either direct access to the content or specific instructions for accessing it.

As indicated previously, the semantic capabilities will be implemented around the topics related to the AV content, treated here as a whole, complete entity. The corresponding field in the DCMI set is the *subject* descriptor, a multi-valued element that stores the relevant topics associated to the content. The role of the ontology will be to function as a controlled vocabulary for the terms that potentially can be assigned to the DCMI *subject* descriptor and capture the semantics relationships of all defined terms. The general architecture is depicted in the figure below:

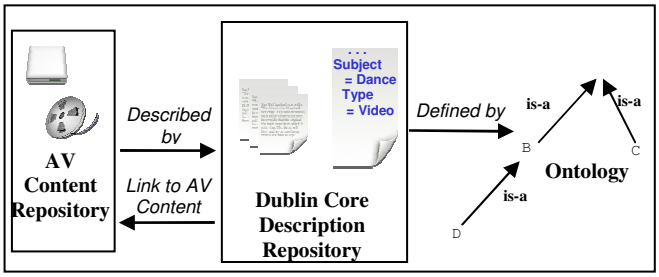


Figure 1. Semantic Searching Architecture

As shown in the figure, applicable values to the *subject* field are defined by the supporting ontology, which also provides the relationships to be explored by the semantic algorithms.

AV content descriptions are inserted into the repository via a cataloguing tool, not depicted in the figure. In general, the description process is initiated by the producer or author of the content. The process consists of textual annotations and requires that all project defined mandatory DCMI fields be filled out in order to allow the retrieval of the content and support the semantic based queries. Semantically enabled fields, such as *subject*, are manually annotated with the aid of specialized end-user interfaces driven by the ontology terms and relationships.

As users access the content descriptions, the catalogued information is improved by means of an established collaboration process. The ontology also evolves collaboratively by means of a cataloguing tool that allows users to suggest new terms and relationships for inclusion in the controlled vocabulary. This contribution will then be analyzed, refined and eventually incorporated into an updated ontology. As a result, the same users that participate in the collaborative content description process may also participate in the improvement of the supporting ontology.

For the development of semantic searching algorithms, it is necessary the combination of metadata descriptions with the supporting ontology [8][9]. Therefore it is important to point out what is required from the structure of the proposed ontology. For this purpose, we will use as example a Brazilian party named *Fandango* [18], a folkloric event in some coastal states, characterized by music and dances that honor sailors and fishermen.

If one catalogs an AV content about *Fandango*, a natural choice would be to set the *subject* field as *Fandango* in the repository. Thus, a semantic enabled searching engine could easily return this item as a response to a query with one of the keywords *Folklore* or *Dance*, as long as the appropriate relationships are present in the ontology. Besides that, if the keywords *Folklore*, *Dance* and *Sailor* were part of the query, the engine could highlight items catalogued as *Fandango* as the ones with the highest probability of matching the actual

 ISBN: 978-1-61208-175-5

interests of the user. Note that non-semantic engines would need to rely on the presence of these keywords in one or more description fields and perform partial match comparisons to get to similar conclusions, thus making the overall process less efficient and error prone.

The question that arises from this example is how the engine will get to such conclusion if the desired item is marked solely as *Fandango*. This is the point where the ontology makes its contribution by providing relationships that make semantic inferences possible. In summary, the ontology needs to be structured in a way that facilitates the categorization of subjects likely to be associated to AV contents [10] and define object properties that establish the semantic relationships between them. Another aspect to be taken into account is to base the ontology on an already established work in order to ease its acceptance by the users. These are key points for elaborating a stable structure of an ontology which can grow in terms of elements and relationships without requiring continuous updates to the deployed software engines. In this project, we selected the controlled vocabulary of the Brazilian Cinemateca [11], a repository of topics for cataloguing contents from independent producers, as the basis for our ontology. This vocabulary is composed of an extensive list of *subjects* which can have one of the following relationships with other *subjects* of the vocabulary:

Subject_A **isA** type of *Subject_B*
Subject_A **isRelatedTo** *Subject_C*

Mapping these relationships to an OWL ontology is straightforward. While **isA** can be directly mapped to the *class<–subclass* or *class<–individuals* OWL relationships, **isRelatedTo** is mapped to an object property whose domain and range are individuals of the generic class *Subject* or any of its subclasses. Note that for this specific application, **isRelatedTo** is not meant to capture the specific aspects that make two given subjects to be related to each other since this would require the definition of an extensive set of properties far beyond what is necessary to accomplish the goals of this work. In our case, the qualified results are obtained by exploring the generic **isRelatedTo** property that may exist between *subjects* defined in the ontology. By applying these relationships to the *Fandango* example mentioned above, we can draw the relationship diagram depicted in Figure 2. As shown in the figure, *subject* derived classes are represented as non-filled rectangles and correspond to groups or categories of topics. Ontology individuals correspond to specialized topics and are represented as solid-filled ellipses. The relationships **isA** and **isRelatedTo** are represented by the solid and dashed arrows respectively. According to the figure, a semantic searching engine could use the following relationships when processing the queries:

Fandango **isA** *Party*, a *Brazilian_Party* and a *Folklore*
Fandango **isRelatedTo** *Sailors* and *Fishermen*

Once the structure of the ontology is defined, we can turn our attention to the proposed semantic algorithms and an illustrative use-case.

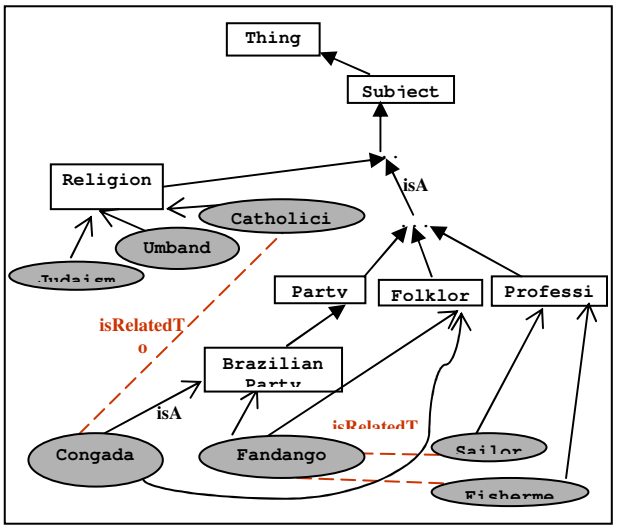


Figure 2. Ontology Structure

IV. SEMANTIC ALGORITHM

In this section we describe the semantic searching algorithm applied in the experiments. The input is the list of query parameters specified by the user for the *subject* descriptor.

Firstly, the algorithm will expand the list of parameters with terms from the ontology having an **isA** relationship with each of the terms entered by the user. This operation expands the list with all sub-classes and individuals members of each parameter entered by the user, thus making the searching process semantically comprehensive. By doing this, any AV content eventually catalogued with one of the specialized terms will also be considered a potential result of the query. The steps of this portion of the algorithm are presented as follows:

(1) ***CTL*** = *Original List of Controlled Terms* – user´s input.
Nc = *number of terms in CTL.*
(2) Let ***i*** be an integer varying from 1 to ***Nc.***
For each term T_i in ***CTL***, create the set S_i defined as:
S_i = {***T,*** (all T_i sub-classes),
(all T_i type individuals)}.
(3) Search the AV contents description repository and select as a result to the query items whose subject descriptor contains values present in all S_i sets.

Note that steps (1) through (3) perform a semantically extended logic AND over the input parameters and furnish a comprehensive list of categories to organize the presentation of the results to the user. For simplicity, we omitted the handling of terms not present in the ontology, which would be treated like any other non-semantic field.

These steps of the algorithm allow a searching engine to return an item marked only as *Fandango* or *Congada* [17] as a possible result for a generic query on the keyword *Brazilian-Party*. Conversely, if the query parameter is the keyword *Fandango*, the searching tool could be enhanced to inform the user that he/she might be also interested in AV contents eventually marked as *Brazilian-Party*. All these

 ISBN: 978-1-61208-175-5

inferences derive from the **isA** semantic relationships defined in the ontology, which were captured in each of the S_i sets created at the step 2. Additionally, for presentation purposes the returned items can be grouped by the elements of each S_i set$_i$, offering to the user a friendlier and more organized interface to navigate into the results of the query.

The benefits of the algorithm are even stronger if we consider that the ontology and the description itself can be evolved as part of a collaborative program. As described in [12], new terms and semantic relationships can be added to the ontology over time and become available to the searching engine by loading the updated ontology into the reasoner. As an example, we can take the term *Fandango*, which according to the ontology, is a type of *Brazilian Party*. However, *Fandango* can also be regarded as a *Brazilian Dance*. As a result of the collaboration process, an updated version of the ontology can specify that *Fandango* is also an individual of *Brazilian-Dance*, a sub-class of *Dance*. As the new ontology is reloaded into the reasoner, *Fandango* annotated items will also be returned for a query on the keyword *Brazilian-Dance*. Note that the results of the query are improved without requiring any updates to the metadata description repository.

The next part of the algorithm aims at obtaining results as close as possible to the actual interests of the user by employing the **isRelatedTo** relationship, as described in the following steps:

(4) Let i be an integer varying from 1 to Nc.
For each term T_i, create the set R_i defined as:
$R_i = \{S_i$ (all individuals which are related to $T_i)\}$.
(5) Let *PITL* be the list of terms present in all R_i sets
PITL is defined as: $\{R_1 \cap R_2 \cap R_3 \cap \ldots \cap R_n\}$.
(6) Include as a qualified result to the query any AV contents whose subject descriptor contains at least one of the terms present in *PITL*.

The short list of qualified results, *PITL*, is represented by the intersection of all the R_i sets, as shown in step (5). If not null, *PITL* contains one or more items common to all elements of the semantically extended lists of terms, the R_i sets, built by exploring the **isA** and **isRelatedTo** relationships over the input parameters. The algorithm infers that *PITL* contains the terms with the highest probabilities of representing the actual intent of the user when submitted the query.

V. USE CASE

The benefits of the semantic algorithm can be better visualized through a practical example in which a description repository contains a couple of instances referencing the term *Fandango* in their *subjects* descriptor fields. Then, let's consider that the user submits a query with the following terms: *Party* and *Folklore*. The execution of steps 1 and 2 will lead to the following:

(1) $CTL = \{Party, Folklore\}$
$Nc = 2$
(2) According to CTL in step 1:
$T_1 = Party$
$T_2 = Folklore$
Now the S_i sets are calculated:
$S_1 = \{Party, Brazilian Party, Congada, Fandango\}$
$S_2 = \{Folklore, Congada, Fandango\}$

At this point, step (3) will return all AV contents whose subject descriptor are marked as *Fandango* since this term is present in both S_1 and S_2 sets. Note that items marked as *Congada* would also be selected as a result to the query, similarly to items marked as *{Brazilian Party*, *Folklore}*.

For illustrating the second part of the algorithm, a slightly different example will be used. The query parameters are now *Party*, *Folklore* and *Sailor* and the goal of semantic query is to obtain AV contents whose associated topics are some how related to all these three terms. The execution of steps (1) thru (2) would lead to the following S_i sets:

$S_1 = \{Party, Brazilian Party, Congada, Fandango\}$
$S_2 = \{Folklore, Congada, Fandango\}$
$S_3 = \{Sailor\}$

Following with the execution of the algorithm, steps (4) thru (6) would lead to:

(4) $R_1 = \{Party, Brazilian Party, Congada, \mathbf{Fandango}\}$
$R_2 = \{Folklore, Congada, \mathbf{Fandango}\}$
$R_3 = \{Sailor, \mathbf{Fandango}\}$
(5) The intersection of all all R_i sets will lead to:
$PITL = \{\mathbf{Fandango}\}$
(6) Now, the engine will search the repository and select AV items marked as *Fandango* as results to the query.

The intersection operation over the R_i sets in step (5) leads to a short list of topics with good probability of representing the real interest of the user. Of course, this depends on the accuracy of the relationships defined in the ontology. In this example, *Fandango* is a subject related to *Sailor* and also a type of *Party* and *Folklore*. Consequently, the algorithm infers that AV contents with *subject* descriptors marked as *Fandango* are the ones with the best chances of meeting the expectation of the user.

It is important to note this conclusion is obtained entirely through inferences made over the supporting ontology. Another benefit of this approach is that as new relationships are added to the ontology, the inference power of the engine increases without requiring updates to the searching engine software.

VI. CONCLUSION AND FOLLOW-ON WORK

The expectation of this work is that the conceived algorithms and ontology structures can effectively contribute for improving the efficiency of searching engines and become a valid mechanism for identifying results more likely to represent the actual interests of the user. At the same time, it is also expected that the algorithms become building blocks for the execution of more complex logical operations involving the entire set of AV contents descriptors fields. At the same time, the availability of semantic enabled searching and cataloguing tools can act as a way to promote the sharing of AV contents in a distributive and collaborative

 ISBN: 978-1-61208-175-5

environment in which both the description and the ontology are continuously improved by the users.

However, validating all these ideas in a real environment is a must. So, the next activities of the project comprise a field test with Brazilian community TV stations and independent AV producers, connected to the cataloguing and searching tools through the GIGA high speed network and the Internet. The diversity of end-users and richness of subjects that can be assigned to AV contents form the ideal combination for establishment of a *de facto* collaboration process where the AV content description and the ontology are gradually refined by the participants.

During the evaluations, we will attempt to test how engines enhanced with semantic capabilities can provide higher levels of effectiveness, accuracy and ease of use when compared to traditional, non-semantic, searching tools. One way to evaluate the proposed algorithm is to define a set of search tasks to be executed by the users, in which some videos must be found. One group of users would then execute searches supported by the proposed semantic algorithm and another group would perform searches in a traditional way. A similar approach, with both quantitative and qualitative results, is presented in [14].

It will be also an opportunity to evolve the structure of the proposed ontology, conceive new algorithms and procedures for the cataloguing and searching process. At the same time, implementation and deployment aspects, such as space-time complexity analysis, scalability issues and performance of the ontology queries will deserve special attention from our research team in order to make sure that the benefits observed in the field experience can be replicated in other environments.

ACKNOWLEDGMENT

This work has been sponsored by the Brazilian Ministry of Communications through the FUNTTEL program and funded by the FINEP innovation agency.

REFERENCES

[1] J. Hunter, “Adding Multimedia to the Semantic Web – Building an MPEG-7 Ontology”, Proc. of the International Semantic Web Working Symposium (SWWS), July 2001, pp. 261–283.

[2] F. Nack and L. Hardman, “Towards a Syntax for Multimedia Semantics”, Technical Report CWI, Amsterdam, The Netherlands, 2002.

[3] J. M. Martínez, R. Koenen, and F. Pereira, “MPEG-7: The Generic Multimedia Content Description Standard”, Multimedia, IEEE, vol. 9(2), 2002, pp. 78-87.

[4] Dublin Core Metadata Initiative, “DCMI Metadata Terms”, http://dublincore.org/documents/dcmi-terms, <retreived: Sept., 2011>.

[5] J. Hunter, “An Application Profile which combines Dublin Core and MPEG-7 Metadata Terms for Simple Video Description”, http://metadata.net/harmony/video_appln_profile.html, <retreived: Sept., 2011>. DSTC- Australia, 2002.

[6] W3C Recommendation, “OWL Web Ontology Language Overview”, http://www.w3.org/TR/owl-features/, <retrieved: Sept., 2011>, February, 2004.

[7] L. Rolim, A. Osorio, and I. Ávila, “Collaborative System for Semantic Annotation of Audiovisual Contents - Applications in the Context of Brazilian Independent Culture” (in portuguese), Proc. of SBSC – Brazillian Symposium of Collaborative Systems, IEEE, 2011, pp. 1-4.

[8] A. Th. Schreiber, B. Dubbeldam, J. Wielemaker, and B. Wielinga, “Ontology-Based Photo Annotation”, University of Amsterdam, IEEE Intelligent Systems, 2001, pp. 66-74.

[9] L. Hollink, S. Little, and J. Hunter, “Evaluating the Application of Semantic Inferencing Rules to Image Annotation”, Proc. of the K-CAP´05 – 3rd International Conference on Knowledge Capture. Alberta, Canada, 2005, pp. 91-98.

[10] A. Isaac and R. Troncy, “Designing and Using an Audio-Visual Description Core Ontology”, Institute National de l´Audiovisuel (INA), France, 2004.

[11] Brazilian Cinematheque, “Controlled Vocabulary”, http://www.cinemateca.com.br/, <retrieved: Sept., 2011>. Audiovisual Department – Brazilian Ministry of Culture, 2011.

[12] H. Klotz and E. P. Wach, “Collaborative Ontology Building”, Seminar Applied Ontology Engineering, STI-Innsbruck – December 2010.

[13] R. Guha, R. McCool, and E. Miller. “Semantic search.” In WWW '03: Proc. of the 12th Int. Conf. on World Wide Web, New York, NY, USA, 2003, pp. 700-709.

[14] .J. Waitelonis, H. Sack, J. Hercher, and Z. Kramer. “Semantically enabled exploratory video search,” Proceedings of the 3rd International Semantic Search Workshop (SEMSEARCH '10), ACM, New York, USA, 2010, pp. 1-8.

[15] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello, “Sindice.com: a document-oriented lookup index for open linked data,” Int. J. Metadata, Semantics and Ontologies. IJMSO, vol. 3(1), 2008, pp. 37-52.

[16] R. W. White, G. Marchionini, and G. Muresan, “Evaluating exploratory search systems: Introduction to special topic issue of information processing and management,” Information Processing & Management, vol. 44 (2), 2008, pp. 433-436.

[17] L. Glaura, “The Sounds of Rosário” (in portuguese), UFMG, Belo Horizonte, 2002.

[18] Caburé Cultural Society, “Live Museum of Fandango” (in portuguese), http://www.museuvivodofandango.com.br/, <retrieved: Sept., 2011>.

 ISBN: 978-1-61208-175-5

Analyzing the Ontology Approaches and the Formation of Open Ontology Model: A Step for Organisational Ontology Employment

Jawahir Che Mustapha Yusuf[1]
Institute of Research and Postgraduate Studies (IRPS)
UniKL-Malaysian Institute of Information Technology
Kuala Lumpur, Malaysia
jawahir@miit.unikl.edu.my

Mazliham Mohd Su'ud
Universiti Kuala Lumpur (UniKL)
UniKL-Malaysia France Institute
Kuala Lumpur, Malaysia
mazliham@unikl.edu.my

Patrice Boursier[1]
Department of Artificial Intelligence, FCSIT
University of Malaya
Kuala Lumpur, Malaysia
patrice@um.edu.my

***Abstract*—Three different ontology-based approaches have been used in previous researches to improve the semantic interoperability in an integrated information system. The approaches can be identified as the single, the multiple and the hybrid ontologies. Organisations seeking to improve their information system capability realise the benefits of using semantic technology based on ontology. However, clear guidelines are not available to select the appropriate ontological approach. The selection of the approach should be according to various organisational needs, contexts and management styles. This research is significantly important to provide flexible and adaptable way to start employing ontology, because semantic information systems are still immature in many organisations. In current research the study of different ontology-based approaches is presented. The focus is on the semantic integration challenge based on multi-sources data integration. Viability of all approaches and guides for ontology employment are presented in order to provide options for the organisations to upgrade their current system to new system. There is no specific approach that has been proven to be a successful implementation. Therefore, a new general reference model is proposed in this research work, which is based on the three approaches called Open Ontology Model. The proposed model is designed to work in dual directions which are top-down and bottom-up implementation to make the specification of ontology mappings more flexible and usable. This model would be of interest to novice system developers who plan to use it as a starting point to develop their first semantic information system. Developers might decide any single or combination of approaches based on the nature of their organisation.**

Keywords-ontology-based information system; semantic heterogeneity; data integration.

I. INTRODUCTION

The needs for knowledge sharing and exchange within organisations have become the most significant and prominent cause of data integration. Therefore, information system interoperability is a key to increase cooperation between all data owners to ensure successful data integration. At present information systems are increasingly large-scale, complex and multi-traits. Information sharing and exchange processes are going to be more challenging. Data integration procedures must follow good abstraction principles to solve interoperability problems concerning on the structure, the syntax, the system and the semantic. The focus of this research is on semantic integration which is one of the main issues in multi-sources data integration.

According to [1], semantic integration is the task of grouping, combining or completing data from different sources by considering explicit and precise data semantics. Semantic integration has to ensure that only data related to the same real-world entity is merged. Ontology is a current practice to resolve semantic conflicts in diverse information sources. Ontology itself is an enabling technology (a layer of the enabling infrastructure) to enforce knowledge sharing and manipulation [2]. Any abstract or concealed information can be clearly described according to specific concepts by using ontology.

Researches to employ ontology approaches for integration of multiple data sources are still growing and more demanding as semantic reconciliation can resolve other types of interoperability problems. Three approaches have been used in previous researches that can be identified as single, multiple and hybrid ontology [9][31]. Large number of systems still holds implicit information even though they might have well support on technical data interoperability. Realizing the growing importance of semantic interoperability, organisations are beginning to use ontologies in their system applications. However, common guidelines to find the ontology approaches that are best suited for different organisational needs, contexts and management styles are still unclear. There are organisations that start with complex approach or approach that is not suitable to some types of organisations. In fact, there exists a

[1] Author is also associated with Laboratoire L3i Université de La Rochelle, La Rochelle, France.

 ISBN: 978-1-61208-175-5

much simpler, cost-effective and quick alternative to be exploited with some improvement. Knowing the advantages and disadvantages of different approaches are not enough to help choose the right approach for a given application. More importantly, there should be a mechanism in place to help the organisations decide the necessary information system upgrades on the basis of their management structure and nature. Furthermore, system developers must deliberately choose proper ontological methods at early stages of system development. Otherwise, invalid result from queried information might yield bogus decision due to poor understanding on the knowledge.

This paper discusses different ontology-based approaches for supporting multi-sources data integration. Viability of all approaches and guides for ontology employment are presented in order to provide options for the organisations to upgrade their current system to new system. A new ontology-based model that is called Open Ontology Model (OOM) is also proposed in this research work. It is intended to be used as general reference model to novice system developers who plan to use it as a starting point to develop their first semantic information system. Developers can take advantage of each ontology approach and may build their systems by stages depends on organisation system requirements and the current resources available. Currently, the prototype of this research work based on the OOM is under implementation.

The rest of this paper is structured as follows: Section II elaborates related researches on ontological-based approaches. Meanwhile, Section III presents the viability of ontological approach and guides to ontology employment. The formation of OOM is detailed out in Section IV. Section V briefs the motivation of this research work. Finally, conclusion is added in Section VI.

II. REVISION ON THE ONTOLOGY APPROACHES

The use of ontologies for data integration is applicable to various numbers of applications. This part describes top-down and bottom-up ontology development. Then, the three ontology approaches based on previous researches contribution in [8][9][10][31] are revised. More recent researches are added to show some earlier approaches still relevant in particular domain background. Indeed, the formation of the Open Ontology Model (OOM) is rooted from the three approaches. The advantages and disadvantages of each approach are not to be emphasized. The concern is more with the numerous types of organisational environments which need to decide the most suitable ontology approaches for their information system upgrade.

A. A Glance on Top-down and Bottom-up Ontology

In computer science perspective, ontology is important for data integration in order to facilitate shared and exchanged information. Generally, two popular trends exist in the development of ontology approaches; top-down and bottom-up designs. In the top-down design, each term in source ontologies is created from the primitive term in the set of top-level ontology. The set of top-level ontology is provided first. Secondly, source ontologies that contain more specific terms are extended from the set of top-level ontology. Since source ontologies only use the vocabulary of a top-level ontology, therefore terms are comparable easily. In the top-level ontology, only common terms are described at a very abstract level. Therefore, adding up existing ontologies should not become a problem as many upper-ontologies (or upper-domain ontologies) are developed under consideration it can be easily reused. The knowledge-base CYC [39], SUMO [38], Sowa's upper ontology [41], WordNet [42], DOLCE [40] and UMBEL [43] are the examples of top-level ontology.

On the other hand, the bottom-up ontology design is aimed to build shared, global ontology by extracting data from source ontologies. Firstly, source ontologies that contain specific terms are constructed from data source schema (or catalogues, labels etc) to describe the meaning of the information. Secondly, source ontologies of all disparate data sources are mapped to construct primitive terms or abstract concepts of the top-level ontology (common shared vocabulary). This way, the related terms between low-level and top-level ontologies are still comparable.

B. Ontology Approaches Revisiting

In *single ontology approach*, a global ontology is derived by data interpretation from all connected data sources as depicted in Fig. 1a. One common shared vocabulary is provided to denote the semantics between data sources. Global ontology development efforts primarily focus on the formation of general knowledge used in multi-purpose applications. A few former systems based on the single approach can be located in the Carnot system [12] that utilises the global CYC ontology [11], an ontology modularization technique in ONTOLINGUA [13], TAMBIS for connecting biological data sources [14], and SIMS [15] as the tightly-coupled system that is tested in the domains of transportation planning and medical trauma. This approach is still utilised in recent years with some improvements such as for spatial data integration in SPIRIT [5][16], a geo-ontology construction for web spatial data query system, three-level ontology architecture for geo-information services discovery in [17] and OCHRE [36] core ontology for combining cultural heritage information from diverse local schemas.

In most real-time implementation, it is not easy to completely achieve mutual agreement within data owners to use one common vocabulary. Thus, m*ultiple ontology approach* is aimed for data integration by mapping different ontologies without using global schema. Each data source is described by its own disparate ontology (Fig. 1b). Inter-ontology mapping technique must be used to enable association between ontologies. Mapping provide a common layer from which several ontologies could be accessed, and hence could exchange information in semantically sound manners [18]. This approach is presented in earlier systems such as OBSERVER system [19] for domain of bibliographic references, combination of two different geographic ontologies using bi-directional integration in [21], MAFRA system [20] and SEWASIE [6]

 ISBN: 978-1-61208-175-5

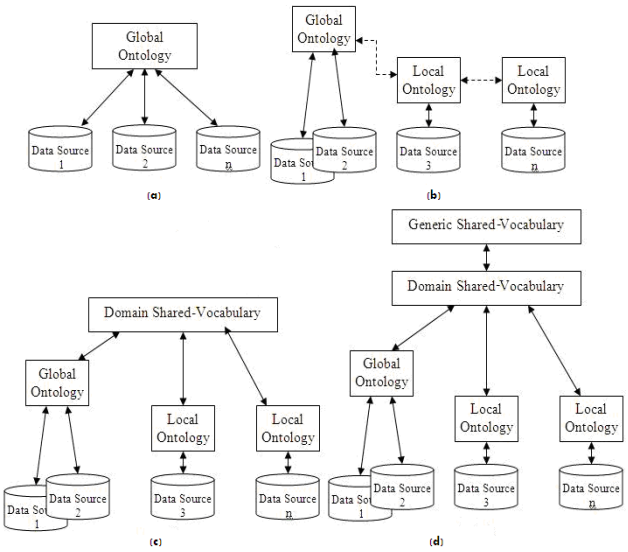


Figure 1. Different ontology approaches: (a) single ontology approach, (b) multiple ontology approach, (c) and (d) two types of hybrid ontology approach that are simulated from [9]

system that use multiple ontologies to provide access to heterogeneous web data and the ontology translation (*bridging axioms)* to merge two related ontologies in OntoMerge XML-based system [4]. More recent work on the approach can be found in [34], where YAGO ontology [33] was automatically derived from Wikipedia and WordNet, further work in [32] to combine high-level axioms from the SUMO and YAGO, and MEMO [35] an automatic merging of two source ontologies, which uses clustering techniques in order to help the identification of the most similar ontologies.

Another mode of multiple ontology integration is done via one shared-vocabulary to make these ontologies simply comparable to each other. This most adopted approach is known as the *hybrid ontology*. Generic ontology and domain ontology are the type of shared-vocabulary. Domain shared-vocabulary can be specified from or without generic shared-vocabulary (Fig. 1c and Fig. 1d). Generic shared-vocabulary usually contains very basic terms in a universe of discourse while domain shared-vocabulary models more specific concept of the world. In some hybrid ontology approach, domain shared-vocabulary is split up into top-domain and domain ontologies as described in [22]. Particularly, hybrid ontology approach set a global top-level ontology to appear as a common reference framework (foundational ontologies) for multi-application and/or multi-domain. The aim is to encourage ontology reuse to facilitate semantic interoperation between applications [10]. At the low-level, all source ontologies that are involved in the integration will use the terms specified in the shared-vocabulary. Simultaneously, each source ontology does not need to be concerned with the context of other source ontologies.

Wache et al. [9] described concisely on the implementation of the hybrid ontology approach in former systems such as COIN, MECOTA and BUSTER. The same approach is used by Elmore et al. [23] to solve a problem of losing data when one global ontology is used. They proposed computer agents over shared-vocabulary to merge only relevant ontologies within participating data sources (USA national lab system). In [3], the authors extended a hybrid ontology approach by defining the XML schema for each data source. The XML schema was then used to create local ontologies before abstracting the equivalent concepts in global ontology. In order to relate between global and local ontology, a mapping rule was applied using path-to-path approach with XQuery language for global query. Bellatreche et al. [24] attempted to achieve a fully automated technique for heterogeneous sources integration of electronic catalogues within engineering databases. Their technique preserves the autonomy of various data sources in which all data sources reference a shared-ontology, and possibly extend it by adding their own concept specializations. In GeoNis [7], semantic mediator was used to solve semantic heterogeneity of geographic data sources. GeoNis provides an ontology mapping between local and top-level ontology, and software support for semantic mismatches. Another related work, GeoMergeP system [25] also created for geographic data sources to focus on the improvement of semantic matching techniques (semantic enrichment and merging).

III. THE VIABILITY OF ONTOLOGY APPROACHES AND GUIDELINES FOR ONTOLOGY EMPLOYMENT

This section justifies the viability of all ontology approaches for different types of organisational environment. Basic guidelines for selecting the appropriate approach in multi-source data integration are also presented.

A. Viability of the Single, the Multiple and the Hybrid Ontology Approaches

In the early generation of ontology-based information systems, data integration adopted the single ontology approach. All data sources should abide with the same agreement to grant a very similar view on the domain. This means all data owners are required to retain and use a single, common ontology definition as well as at it local schema. Single ontology environment depicts that the newly added data source is modelled using terms from general, shared domain model only. Furthermore, a global ontology is also possible to be extended if the new data source goes beyond what is modelled in the current global ontology. Any changes such as alteration and deletion in data source will also imply the changes in global ontology. However, all the tasks are bounded by the size of the required data sources.

The integrated system based on the single ontology approach is applicable to certain environments which comply with specific principles. The single ontology mechanism is fine if data sources schema have no pre-existing ontologies and at once agreeable to use a global vocabulary. Data integration could be done if all data sources are able to share similar view on a domain of interest. The former mechanism

(i.e., SIMS), if changes occur in any data sources, will affect current global ontology and their mappings with other data sources. In order to resolve this issue, the creation of a mapping rule such as in [3] between a global ontology and local schema could be applied. Therefore, new sources can easily be added without the need to use a global ontology modification but only the mapping rule. Integration method in [37] is also feasible because the authors created user ontology that was independent of databases and similarity functions to compare related entities and instances in the system. User ontology allows users to express queries in their own terms according to their own conceptualizations without having to know the underlying modeling and representation of data in heterogeneous databases. Any updates in both the user ontology and the databases will not affect the system. Another issue of using this approach is the possibility to lost a valuable concepts of information could happen as described in [23]. If two or more data sources do not have a common view on some prospective information, it will not be appended in global ontology. This issue can still be resolved if some uncommon concepts are critically decided upon to be a sharable concepts in global ontology.

In other perspective, this approach is hard to support due to the complexities involved in integrating the ontologies and maintaining consistency across concepts from different ontologies with only a single shared-vocabulary [19]. On top of that, data sources should have full autonomy to sustain its own datasets. Thus, this approach is possible to be applied in less distributed environments where only fewer data sources exist and this situation enables simple ontology mapping process to be done. In a less heterogeneous organisational model such as in intra-government agencies, this approach can also be considered. Additionally, the frequency of future changes also should be nominal to avoid complexities while maintaining the integrated system. Overall, when the principles in single ontology approaches are difficult to be attained an alternative ontology approaches could be considered.

In the multiple ontology approach, the tasks such as insertion, exclusion or alteration of data sources are easily supported. Each data source has its own autonomy without being dependent on a global schema. The correlation between pre-existing multiple ontologies is easier than creating a global ontology because a smaller community is involved in the mapping process [20]. SEWASIE [6] developer also claimed that at the local level, things may be done more richly than at a wider level. In contrast, to compare different ontology sources are more challenging without common vocabulary. Furthermore, inter-ontology mapping is also prone to the complexities in query process. Although the use of inter-ontology mapping in [20] and [6] are rational, but system developers must also be concerned with the integration of large different ontologies. We might involve more complicated tasks of creating multiple mapping processes if existing mapping rules cannot be applied directly on new local ontology. Otherwise, this approach is simple and feasible.

Inter-ontology mapping is actually quite challenging to define in the environment when more than two information sources exist in the domain of interest. Mapping tasks become more complex as system developers might discover more semantic heterogeneity problems to correlate the ontologies between all the multiple sources. Many other mapping techniques are not clearly defined [26] and still remain as a research attention over recent years. Some discussions upon mapping for multiple ontology approaches can be referred at [26][27]. In other point of fact, the integration of a particular type of information within geographic and non-geographic data encompasses excellent implementation when using this approach, for instance in the domain of disaster management, forestry, land planning, and agriculture just to name a few. These kinds of information are typically distinct and independent in nature, and also in its description. They usually contain at least one common concept that could be related to strengthen the meanings of information. Thus, promising for data integration to facilitate effective information sharing under specific domain.

Data sources autonomy is partially vanished in former systems, which were based on the hybrid ontology approach. The existing ontologies cannot easily be reused and need to be redeveloped from scratch [9] by referring to the shared-vocabulary. Path to path approach and abstraction method as used in [3], and Ontology-based Database (OBDB) approach introduced in [24] could resolve the problem because the newly added data source is still able to maintain the autonomy by using its own local concepts. The hybrid ontology is a well-known approach that allows new data sources to be added easily in the ontology-based system. If new data source contains concepts that are not described with ontologies, local ontologies will be created for it by referring to the general terms established in shared-vocabulary. The sharable terms which are not specified in shared-vocabulary will be added directly in shared-vocabulary as general terms. Then, the mapping process of new terms is created to relate between local and shared-vocabulary. If new data sources come with pre-existing ontologies, system developer should investigate whether shared-vocabulary (upper to very upper level) is present or not. With the existence of shared-vocabulary, the different source ontologies should refer to the upper ontology with liberty to preserve its own concepts. The source ontologies may extend the upper ontology as much as required. Without shared-vocabulary, the different source ontologies could be connected using bottom-up direction to produce it common terms. The global ontology as in the single and the hybrid ontology approaches are actually transfers the burden of information correlation and filtering on the query processing system [19]. With global shared-vocabulary, the integration of pre-existing ontologies using global-local mapping rules will lessen the complexities in creating the query process compared with inter-ontology mapping.

B. A Proposed Guidelines for Ontology Employmet

Ontology-based information system for organisations (public and corporate sector) is still an immature field. Readiness for change to apply a formal ontological approach is a key factor to successful modern application integration solution. The selection of appropriate ontology approach is

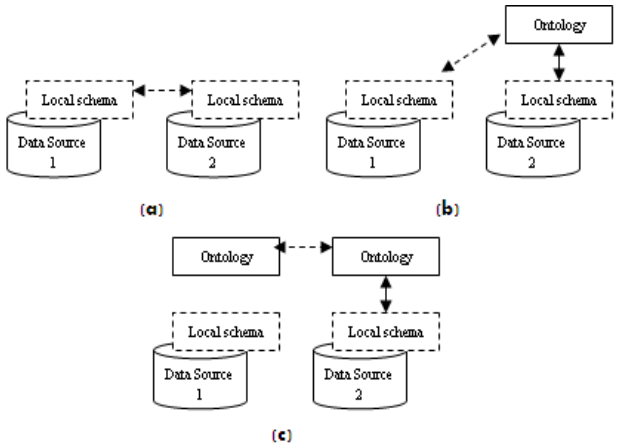


Figure 2. Integration of various system's structure: (a) Less explicit systems, (b) less explicit system and ontology-based system, (c) multiple ontology-based systems

solely depended on the organisational environment. Although the hybrid ontology perform well in most situations, the single and the multiple ontologies are also practical. Both approaches offer fast, economical and can possibly to be extended to the hybrid ontology. Once the data owners agree to use the ontology, they must properly decide on the ontology approach which is suitable for their organisation. So, organisational nature, scopes, information needs and resources are important in selecting the practical approach for ontology-based multi-source data integration.

Obviously, the majority of the current system holds less explicit information system integration (Fig. 2a). Modern information system is encouraged to embed more semantics in their systems to allow better information integration and this could be achieved by using ontology. Based on Section III-A, the single ontology approach is recommended if the data owners and their system conform to the following states:

1. Each data source contains at least one common concept and some uncommon concepts are declared sharable in global ontology to avoid data loss.
2. Each data owners participating in the integration process agree to use similar definition of global ontology.

Small-scale enterprise and intra-agencies usually possess common datasets that are maintained in distributed location. The single ontology will be practical for them in order to achieve low-cost, low-risk and fast deployment of semantic-based integrated system. The multiple ontology approach works very well if only two data sources are involved in the integration. Otherwise, hybrid ontology approach is more convenient as mapping process beneath global ontology simplify the complexities in inter-ontology mapping. In order to develop their first ontologies, data heterogeneities will be the first problem faced by the developers. Many research such as in [3][4][7][17][23][24] gave solutions to reconcile the heterogeneities.

In another situation, a possible integration could occur between less explicit data source with an ontology-based system (Fig. 2b). The first problem is to match local schema with pre-existing ontology. There is a possibility to reuse existing ontology as a global ontology (single ontology approach) if each data sources is able to share similar concepts. Otherwise, new ontology for non-ontology-based data source could be developed to enable peer-to-peer or hybrid ontology integration.

More challenges would be face by the system developers to integrate multiple ontologies (Fig. 2c). The problem here is the ontology heterogeneity. Even if each data source has its own ontology, the heterogeneity problems will still not resolved. Ontology merging is a common approach to combine existing ontology into common vocabulary that incorporates possible aspects of participating ontologies [27]. Another way to integrate multiple ontologies is thru ontology matching in order to define equivalent relation between different ontologies. The system developers should be able to resolve the inter-ontology integration complexities and maintaining consistency across different concepts. Euzenat and Shvaiko [28] described in detail how the matching technique should work for multiple ontologies. Even though having few complexities along with high cost and long-time implementation, the hybrid ontology approach could work well with pre-existing ontologies.

With regards to the selection of ontology approaches single ontology approaches will never suit with sustained and entrenched organisational models due to its costly transformation and maintenance process. Multiple ontology approaches is feasible if the developer is able to maintain all ontologies. They might create inter-ontology mapping (traversing semantic relationship) via terminological relationship. Less complexity in inter-ontology mapping can be achieved if ontologies which are to be integrated are nominal. Thus, this approach is not recommended for huge number of different specific ontologies as it becomes a great effort to traverse and understand all the semantic relationship. As such, the hybrid ontology approach that is supported with broad mapping techniques can almost fit all environments.

A notion that could add little add-ons to the organization ontology modelling theory is presented: Even though ontology is to describe the explicit meaning of knowledge, there is no explicit or better approaches for ontology employment since it really depends on the organisational structure and its management style, in accord with their scopes, the type of external information needs, and also the available resources such as personnel, financial, physical and their internal information itself.

IV. THE FORMATION OF OPEN ONTOLOGY MODEL

OOM (Fig. 3) is a general reference model for organisations data integration at semantic level. This model is meant for various domains of application (i.e., E-Government, Crisis Management etc.), to interconnect multi-sources data particularly on database components. Ontology building is expected to work in dual directions; top-down and bottom-up implementation. The model is aimed to be a flexible model for ontology employment by the organisations. The ontology-based model should in principle adopt a general to specific approach. Thus, the model is

 ISBN: 978-1-61208-175-5

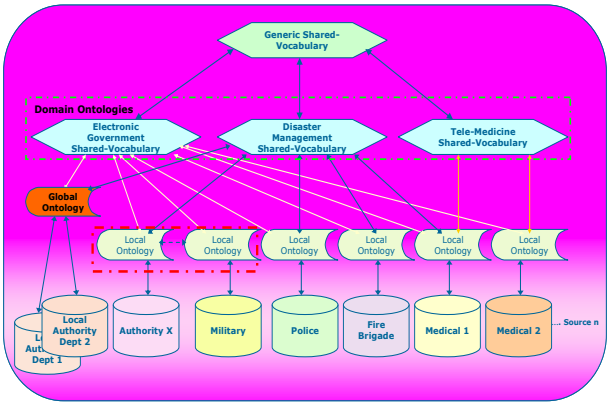


Figure 3. The Open Ontology Model

adequately expansive for explicit semantic data integration to avoid potential problems of under-specification. Afterward, the organisations can legitimately simplify the model according to their management needs. Obviously, the OOM is designed as the combination of available ontology approaches that feasible in most organizations environment.

In this model, each classes and property is assigned with *primary identifier* as in PLIB ontology [29] to map between concepts. The model approach works with or without existing source ontologies. It is assumed that generic or domain shared-vocabulary exists to be referred by low-level ontology (top-down to bottom-up). But it doesn't mean that explicit mapping correlation must be made to refer to the upper ontology. This happens when the participating organizations decide to use the single ontology approach. The single ontology is constructed with consideration on the existence of the upper ontology, so that the single ontology will be constantly ready for upgrading into hybrid ontology for connecting multiple data sources. That is also similar with the organisations who decide to use the multiple ontology approach. Two participating data sources shall contain its own ontology that is created in advance with respect that there exist a generic or domain shared-vocabulary. In future, mapping rules to connect between two ontologies may be used to adapt with hybrid ontology environment.

Hybrid ontology approach is anytime viable to associate less or more data sources. If the participating data sources in the integration process have no pre-existing ontologies, each local ontology will be created with reference to shared-vocabulary. The local ontology possibly will extend its body to have more specific entities and properties. In the pre-existence of ontology, this source still has the autonomy to maintain its name concepts. The *primary identifier* is used to indicate the similarity or different concepts between participating data sources and it upper-vocabularies. Fig. 4 depicts the top-down to bottom-up mapping implementation with the use of *primary identifier*.

Local ontology is defined based on the schema of the local database. Data owners will decide their own definition of local ontology concepts. Concepts that are rational to be disclosed will be pulled out to domain-shared list. Concealed concepts (shaded in Fig. 4) will not be shared but can be

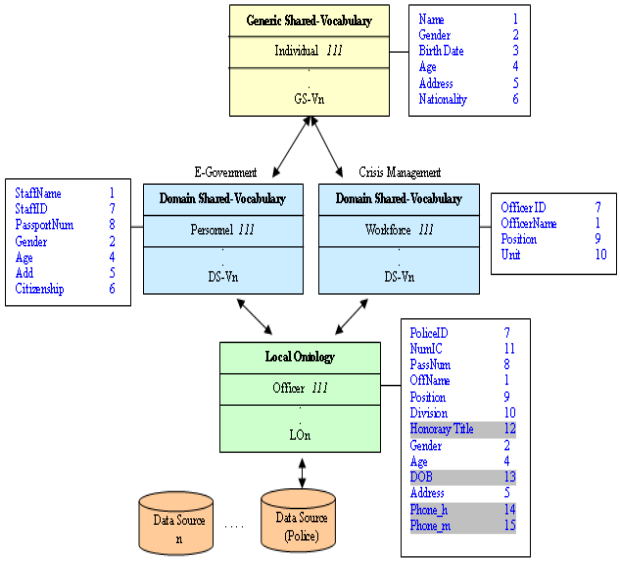


Figure 4. Top-down to bottom-up mapping

accessed locally or may be shared (right away or later) in different domain. Generic and domain shared-vocabulary are the list of shared concepts for all participating data sources. In our approach, the design of shared-vocabulary begins with inspirational approach [30]. For instance, 'National Security Division' as the principal initiates the specification of generic and domain shared-vocabulary that is substantially potential to be shared with the group of the data owners. Concerned with the importance of information sharing, the data owners may collaboratively [30] use the existing shared-vocabulary as the anchor and supportively extend it if necessary. However, the data source owners will not be attentive to each other's data. This is important for most of the intelligence systems that are confidentiality-related. Some ontology standards (ISOs, ANSI etc.) and/or other common top-level ontologies (WordNet, OpenCyc etc.) may be reused during the ontology design time.

V. MOTIVATION OF THE RESEARCH WORK

The prototype of research work based on the OOM is currently under implementation. The attention is given to perform an ontology-based integrated system beneath the crisis management domain within the Malaysian public agencies, particularly amongst local authorities, police, fire brigades and medical agencies. An example of study is drawn from digitized, multi-format documents that are collected before and after disasters. The data sets are typically stored in heterogeneous GIS-based (raster images or vector) proprietary or open formats such as Shapefile, MapInfo TAB, GML, KML JPEG2000, DEM, GeoTIFF, etc. Besides, some photographic images, text-documents, video and audio clips which are collected aftermath of a disasters allows the decision makers to see the big picture of the disaster events. Even though they are maintained and distributed by different information systems, formats, organizations and locations, but their contents might carry

 ISBN: 978-1-61208-175-5

one and the same calamity story, situation, related and supporting each other. Access to all of this valuable data needs high performance of information retrieval and integration mechanism that is effective at gathering, analyzing and outputting the required information.

Malaysia has good mechanism in managing disasters and the committee was established at three different levels (Federal, State and District under National Security Division Secretariat) to coordinate all the activities related to disaster. Various agencies perform their own daily work routine and maintain their own information either manually or in digitized form (flat files, databases and etc.). During disaster events, huge amount of information are acquired to be disseminated amongst them. However, the required datasets are not only difficult to obtain from system network but lack of automated data coordination at operational level such as during counter-disaster, rescue and relief activities. In addition, if information system is utilised, each agency may use different terminology to refer to similar data, and different document format to store spatially and semantically related information. Ontology usage in information system is still at infant stage amongst the Malaysian public agencies. Furthermore, ontology in this domain is not yet exists in the context of Malaysian disaster management. This research opens up significant opportunities to achieve more flexible and adaptable way to start employing ontology within many organisations.

VI. CONCLUSION AND FUTURE WORKS

Various ontology-driven information system approaches for multi-sources data integration is presented to provide direction for ontology employment among different organizations. Based on this study, the organizations should not adhere to employ directly specific model approach but are given as much autonomy as possible with respect to their nature along with their resource allocation and acquisition. Both the single and the multiple ontologies have high level of implementation feasibility because the approaches provide a quick way to develop quick, low risk and low-cost system application. Furthermore, the approaches may be extended to hybrid ontology when greater integration of heterogeneous data sources is required. A hybrid ontology approach can almost fit all environments but the challenge of having more ontology heterogeneity could delay the development. Besides a flexible OOM that is feasible in most organizations environment is also proposed. The ontology is designed to follow inspirational and collaborative approach with the top-down to bottom-up implementation. The OOM could be replicated in developing the semantic-based application for various domains of interest.

The presented model approach to design an ontology provides the basis for developing and implementing the ontology-based system. The system is aimed to improve multi-source, multi-format document query and integration particularly for disaster management domain. Further research is focusing to make better the ontology building, along with testing and evaluating the concepts in domain and application ontology. The ontology matchmaking is primarily come into focus to help achieve the goal of automatic data search and integration to response a specific query.

ACKNOWLEDGMENT

We would like to express our sincere thanks to IRPS-UniKL and Majlis Amanah Rakyat (MARA) for the research funding and administrative support, Laboratoire L3i Université de La Rochelle for research and administrative support, and we are also thankful to Mr. Muhammad Alam and Ms. Kim de Silva for advising us on this paper.

REFERENCES

[1] P. Ziegler and K. R. Dittrich, "Three Decades of Data Integration - All Problems Solved?," in 18th IFIP World Computer Congress (WCC 2004), Building the Information Society, vol. 12, Aug 2004, pp. 3-12.

[2] T. R. Gruber, "Ontology of Folksonomy: a Mash-up of Apples and Oranges," International Journal on Semantic Web and Information Systems, vol. 3(1), 2007, pp. 1-11.

[3] L. Zhang, Y. Ma, and G. Wang, "An Extended Hybrid Ontology Approach to Data Integration," in Proceedings of 2nd International Conference on Biomedical Engineering and Informatics (BMEI '09), Tianjin, China, 2009, pp. 1-4.

[4] D. Dou, D. McDermott, and P. Qi, "Ontology Translation on the Semantic Web," On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, 2003. pp. 952-969.

[5] C. B. Jones, A. I. Abdelmoty, D. Finch, G. Fu and S. Vaid, "The Spirit Spatial Search Engine: architecture, ontologies and spatial indexing," Geographic Information Science, 2004, pp. 125-139.

[6] S. Bergamaschi, F. Guerra, and M. Vincini, "A peer-to-peer information system for the Semantic Web," Agents and Peer-to-Peer Computing, 2005, pp. 161-182.

[7] L. Stoimenov, A. Stanimirovi, and S. Djordjevic-Kajan, "Semantic Interoperability using multiple ontologies," Proceedings printed as book, Eds. Fred Toppen, Marco Painho, AGILE, 2005, pp. 26-35.

[8] U. Visser, H. Stuckenschmidt, and C. Schlieder, "Interoperability in GIS-enabling technologies," in Proceedings of the 5th AGILE Conference on Geographic Information Science, Palma de Mallorca, Spain, 2002, pp. 25-27.

[9] H. Wache, et al., "Ontology-based integration of information - a survey of existing approaches," in Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, 2001, pp. 108-117.

[10] N. F. Noy, "Semantic integration: a survey of ontology-based approaches," ACM Sigmod Record, vol. 33, 2004, pp. 65-70.

[11] D. Lenat and R. V. Guha, Building Large Knowledge-based Systems: Representation and Inference in The Cyc Project: Addison-Wesley, 1990.

[12] C. Collet, M. N. Huhns, and W. M. Shen, "Resource integration using a large knowledge base in Carnot," IEEE Computer, 1991, pp. 55-62.

[13] T. R. Gruber, "Ontolingua: a mechanism to support portable ontologies," Knowledge Systems Laboratory, Stanford University, California, Tech. Rep. Version 3, 1992.

[14] R. Stevens, et al., "TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources," Bioinformatics, Oxford Univ. Press, England, vol. 16(2), 2000, pp. 184-185.

[15] Y. Arens, C. A. Knoblock, and W. M. Shen, "Query reformulation for dynamic information integration," Journal of Intelligent Information Systems, vol. 6, 1996, pp. 99-130.

 ISBN: 978-1-61208-175-5

[16] G. Fu, C. B. Jones, and A. I. Abdelmoty, "Building a geographical ontology for intelligent spatial search on the web," in Proceedings of International Conference on Databases and Applications, 2005, pp. 167-172.

[17] F. Probst and M. Lutz, "Giving meaning to GI web service descriptions," in 2nd International Workshop on Web Services: Modeling, Architecture and Infrastructure (WSMAI), Porto, Portugal, INSTICC Press, 2004.

[18] Y. Kalfoglou and M. Schorlemmer, "Ontology mapping: the state of the art," The Knowledge Engineering Review, vol. 18, 2003, pp. 1-31.

[19] E. Mena, V. Kashyap, A. Sheth and A. Illarramendi, "OBSERVER: an approach for query processing in global information systems based on interoperation across pre-existing ontologies," Distributed and Parallel Databases, vol. 8(2), 2000, pp. 223-271.

[20] A. Maedche, B. Motik, N. Silva and R. Volz, "MAFRA - A MApping FRAmework for distributed ontologies," Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, 2002, pp. 69-75.

[21] F. T. Fonseca and M. J. Egenhofer, "Ontology-driven Geographic Information Systems," in 7th ACM Symposium on Advances in Geographic Information Systems, Kansas City, MO, November 1999, pp. 14-19.

[22] H. Stenzhorn, E. Beisswanger, and S. Schulz, "Towards a top-domain ontology for linking biomedical ontologies," Studies in Health Technology and Informatics, vol. 129(2), 2007, p. 1225-1229.

[23] M. Elmore, T. Potok, and F. Sheldon, "Dynamic Data Fusion using an Ontology-based Software Agent System," Proceedings of the IIIS Agent Based Computing, Orlando, vol. 7, 2003.

[24] L. Bellatreche, N. X. Dung, G. Pierra and D. Hondjack, "Contribution of Ontology-based Data Modeling to Automatic Integration of Electronic Catalogues within Engineering Databases," Computers in Industry, vol. 57, 2006, pp. 711-724.

[25] A. Buccella, et al., "GeoMergeP: Geographic information integration through enriched ontology matching," New Generation Computing, vol. 28, 2010, pp. 41-71.

[26] S. M. Falconer, N. F. Noy, and M. A. Storey, "Ontology mapping - A user survey," in Proceedings of The Second International Workshop on Ontology Matching, 2007.

[27] J. de Bruijn, et al., "Ontology mediation, merging and aligning," in Semantic Web Technologies: Trends Research and Ontology-based Systems, J. Davies, R. Studer and P. Warren, Eds. Chichester, UK: John Wiley and Son's, 2006, pp. 95-113.

[28] J. Euzenat and P. Shvaiko, "Ontology matching," Springer-Verlag, New York Inc., 2007.

[29] G. Pierra, "The PLIB ontology-based approach to data integration," in Proceedings of the 18th IFIP World Computer Congress (WCC'2004), Toulouse France, 2004, pp. 13-18.

[30] C. W. Holsapple and K. D. Joshi, "A collaborative approach to ontology design," Communications of the ACM, vol. 45, 2002, pp. 42-47.

[31] U. Visser, Intelligent Information Integration for the Semantic Web. Springer-Verlag, New York Inc, 2004.

[32] G. De Melo, F. Suchanek, and A. Pease, "Integrating YAGO into the Suggested Upper Merged Ontology," in Proceedings of 20th IEEE International Conference on Tools with Artificial Intelligence, IEEE Press, 2008, pp. 190-193.

[33] F. M. Suchanek, G. Kasneci, and G.Weikum. "Yago: A core of Semantic Knowledge". in Proceedings of World Wide Web, New York, USA, ACM Press, 2007, pp. 697-706.

[34] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A large ontology from Wikipedia and Wordnet," Journal of Web Semantics: Science, Services and Agents on the World Wide Web, vol 6(3), 2008, pp. 203-217.

[35] F. F. de Araújo, F. L. R. Lopes, and B. F. Lóscio, "MeMO: A Clustering-based approach for merging multiple ontologies," Workshops on Database and Expert Systems Applications, Bilbao, Spain, IEEE Press, 2010, pp. 176-180.

[36] Online Cultural Research Heritage Envirenment. (2011, May 8). Core Ontology. [Online]. Available: http://ochre.lib.uchicago.edu/index_files/Page845.htm

[37] M. Gutiérrez and A. Rodríguez, "Querying heterogeneous spatial databases: Combining an ontology with similarity functions," in Proceedings of the ER Workshop on Conceptual Modeling of GIS, 2004, pp. 160–171.

[38] A. Pease, I. Niles, and J. Li, "The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications," in Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web, Edmonton, Canada, 2002.

[39] C. Matuszek, J. Cabral, M. Witbrock, and J. DeOliveira, "An introduction to the syntax and content of Cyc," in Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and its Applications to Knowledge Representation and Question Answering, 2006, pp. 44-49.

[40] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider, "Sweetening ontologies with DOLCE, Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, 2002, pp. 223-233.

[41] J. F. Sowa, Knowledge Representation - Logical, Philosophical, and Computational Foundations, Pasific Grove, California, USA, Brooks Cole Pulishing Co., 2000.

[42] Princeton University. (2011, June 21), WordNet A Lexical Database for English. [Online]. Available: http://wordnet.princeton.edu/wordnet/

[43] M. K. Bergman, and F. Giasson. (2008, July), Umbel Ontology-Subject Concepts and Named Entities Instantiation, Technical Report, vol. 2. [Online]. Available: http://fgiasson.com/articles/UMBELOntology_vA2.pdf

 ISBN: 978-1-61208-175-5

An Approach for Indexation, Classification and Retrieval of Knowledge Sources in Collaborative Environments

Ruben Costa
Centre of Technology and Systems
UNINOVA
Quinta da Torre, Portugal
rddc@uninova.pt

Celson Lima
Federal University of Western Pará
UFOPA / IEG / PSI
Santarém, Brasil
celsonlima@ufpa.br

***Abstract*—This work introduces a conceptual framework and its current implementation to support the semantic enrichment of knowledge sources. It improves the ability for indexing and searching of knowledge sources, enabled by a reference ontology and a set of services which implement the searching and indexing capabilities. Essentially, our approach defines an appropriate knowledge representation based on semantic vectors which are created using three different but complementary algorithms for each knowledge source, using respectively the concepts and their equivalent terms, the taxonomical relations, and ontological relations. We introduce the conceptual framework, its technical architecture (and respective implementation) supporting a modular set of semantic services based on individual collaboration in a project-based environment (for Building & Construction sector). The main elements defined by the architecture are an ontology (to encapsulate human knowledge), a set of web services to support the management of the ontology and adequate handling of knowledge providing search/indexing capabilities (through statistical/semantically calculus). This paper also provides some examples detailing the indexation process of knowledge sources, adopting two distinct algorithms: "Lexical Entries-based" and "Taxonomy-based". Results achieved so far and future goals pursued here are also presented.**

Keywords-Knowledge Engineering; Ontologies; Indexation; Classification; Retrieval

I. INTRODUCTION

Over the last two decades, the adoption of the Internet as the primary communication channel for business purposes brought new requirements especially considering the collaboration centred on engineering projects. Engineering companies are project oriented and successful projects are their way to keep market share as well as to conquer new ones. From the organisation point of view, knowledge goes through a spiral cycle, as presented by Nonaka and Takeuchi in the SECI model [1]. It is created and nurtured in a continuous flow of conversion, sharing, combination, and dissemination, where all the aspects and dimensions of a given organisation, are considered, such as individuals, communities, and projects.

Knowledge is considered the key asset of modern organisations and, as such, industry and academia have been working to provide the appropriate support to leverage on this asset [2]. Few examples of this work are: the extensive work on knowledge models and knowledge management tools, the rise of the so-called knowledge engineering area, the myriad of projects around 'controlled vocabularies' (e.g., ontology, taxonomies, thesaurus), and the academic offer of knowledge-centred courses (graduation, master, doctoral).

As relevant literature shows [3]; [4]; [5]; [6], knowledge management (KM) does not only comprise creation, sharing, and acquisition of knowledge, but also classification, indexation, and retrieval mechanisms (see Figure 1). Knowledge may be classified by its semantic relevance and context within a given environment (such as the organisation itself or a collaborative workspace). This is particularly useful to: (i) improve collaboration between different parties at different stages of a given project life cycle; and (ii) assure that relevant knowledge is properly capitalised in similar situations. For example, similar projects can be conducted in a continuously improved way if lessons learned from previous are promptly known when a new (and similar to some previous one) project is about to begin.

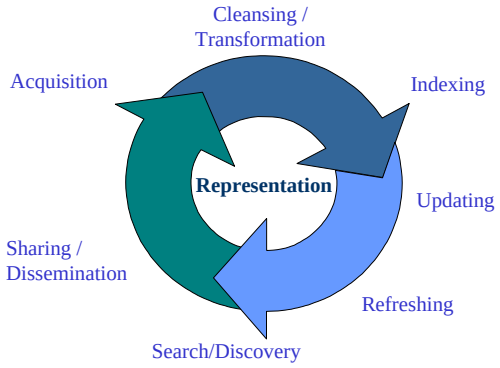


Figure 1. KM Lifecycle

Semantic systems utilize an ontology (or a set of ontologies) to encapsulate and manage the collection and representation of relevant knowledge, hence giving information a human-relevant meaning. Semantic description of project resources enhances collaboration through better understanding of document contents (supporting better understanding and extraction of knowledge) [7]. In addition, by introducing ontological reasoning, semantic techniques enable discovery of knowledge and information that was not part of the original use case or purpose of the ontology itself [8].

 ISBN: 978-1-61208-175-5

The work presented here provides project teams with semantic-enabled services, targeting the improvement of the semantic richness of knowledge sources (KS) used/created, during the execution of an engineering project. The work conceptually covers two dimensions, namely collaboration and knowledge engineering, focused on ontology development and knowledge sharing activities [9]. Knowledge, the dimension particularly explored in this paper, relates to the 'currency' being exchanged during a collaborative process, in this case a collaborative engineering process. Technical documents, lessons learned, and expertise, are some examples of such currency.

This paper is structured as follows. Section 2 defines the objectives and addresses the problem to be tackled. Section 3 introduces the software components handling the knowledge related matters previously introduced. Section 4 gives illustrative examples of the software operation. Section 5 explains the need for conducting more empirical results. Finally, section 6 concludes the paper and points out the future work to be carried out.

II. RELATED WORK

Index terms are traditionally used to characterize and describe the semantics of a document. Such approach attempts to summarize a whole document with a set of terms that are relevant in the context of the document. While this approach has given some satisfactory results in the area of Information Retrieval (IR), it still has some limitations as it proceeds by oversimplifying the summarization process by relying on a subset of relevant terms that occur in a document, and uses these as a mean to convey the semantics of the document. The most commonly used IR models are: Boolean, Vector and Probabilistic [14]. In the Boolean model, documents are represented as a set of index terms. This model is said to be set theoretic [15]. In the Vector model, documents are represented as vectors in a t-dimensional space. The model is therefore said to be algebraic. In the probabilistic model, the modelling of documents is based on probability theory. The model is therefore said to be probabilistic. Alternative models that extend some of these classical models have been developed recently. The Fuzzy and the Extended Boolean Model have been proposed as alternatives to the set theoretic model. The Generalized Vector, the Latent Semantic Indexing, and the Neural Network models have been proposed as alternatives to the Algebraic Model. The Inference Network, and the Belief Network models have been proposed as an alternative to the Probabilistic Model.

It is also worth mentioning that models that reference the structure, as opposed to the text, of a document do exist. Two models have emerged in this area: the Non-Overlapping Lists model [16] and the Proximal Node model [17]. Our approach enhances the vector-space model for IR, by adopting an ontology based implementation. It implements the notion of semantic vectors, which takes into account the taxonomical and ontological relations between concepts, which is an aspect that is neglected by most of IR approaches nowadays.

The e-cognos project [12] addressed this issue, but its major outputs remain only at a first level of IR, described in this work as lexical entries based indexation. A more recent work also addresses this theme, by enhancing the vector space-model [13], but it does not take into account the ontological and taxonomical relations of ontology concepts, adopting a different approach as the one presented in this work.

III. RELEVANCE OF THE WORK

The key question guiding the development of this work is: How to augment the relevance of knowledge sources in collaborative engineering projects in order to support users within problem-solving interactions?

The traditional method of turning data into knowledge relies on manual analysis and interpretation. For example, in the building & construction domain, it is common for specialists to periodically conduct several simulations before start building, on a regular basis. The specialists then provide a report detailing the analysis to the building owners and building contractors organizations; this report becomes the basis for future decision making and planning for building & construction.

This form of manual probing of a data set is slow, expensive, and highly subjective. In fact, as data volumes grow dramatically, this type of manual data analysis is becoming completely impractical in many domains. Who could be expected to digest millions of records, each having tens or hundreds of fields? We believe that this job is certainly not one for humans; hence, analysis work needs to be automated, at least partially.

On the other hand, systems are normally focused on the management of structured information, but they also include a wide range of unstructured information in the form of documents, drawings, images, etc.. Thus, although there might be an understood relationship between a document and a part of the product structure, there are still concerns about how to more effectively make the information and knowledge stored in such systems available to and useful for a wide range of actors in collaborative environments.

In comparison to structured information, the unstructured information lacks context, and since there are no predetermined data types or established relationships between dispersed pieces of information, it is often difficult to find such information if you do not know exactly what you are looking for. For example, when searching for documentation of a certain decision, it might be needed to browse through a vast amount of e-mail, meeting notes, spreadsheets, or blog posts, and the only help available is usually a free-text search that does not always return relevant results. In the specific case of documents, it is often to find metadata in the form of the file name, the date it was created, the version history, the name of the person who created the document, but this information usually says little about the relevance and usefulness of the actual content.

It is important to highlight that a document, or any other kind of unstructured piece of information that has been stored in a database, does not mean that the content is easily

 ISBN: 978-1-61208-175-5

retrieved or analysed beyond the individual or team that took part in the creation of the document.

A. *Objectives*

The main objective pursued here is related with capturing and reuse of knowledge, by adopting an ontology-based approach using semantic and statistical/proximity based algorithms to better augment the relevance of knowledge sources created/used within collaborative engineering projects. In this sense, the key capabilities to be provided are the following:

- Knowledge documentation and storage: support a consistent approach for documenting lessons learned in ontology-based system that allows semantic retrieval of documents.
- Knowledge classification: knowledge classification is a highly desirable functionality and one having a high priority. Existing tools only allow for the categorisation of knowledge. It is more important to support knowledge item clustering (finding similarities between knowledge items).
- Search for knowledge items: the search, discovery, and ranking of knowledge items are issues of high priority with respect to both the manner in which these are done and in terms of the different types of knowledge items considered (full text search; searches on the basis; and discovery of experts and communities).

This work aims to provide the best ontological representation for a given knowledge source within a given context, when adding/searching for knowledge. When adding a new knowledge into the knowledge repository, the approach being implemented will extract the best relevant keywords from the KS and calculates their statistical weights. This set of keywords/weights forms the basis of the so-called Semantic Vector (SV), which is analysed against the ontology in order to get the ontological representation of the KS, which is defined by concepts from the ontology. A knowledge representation is then built for the KS and stored into the repository. This is going to be explained more clearly in the following sections.

When searching for knowledge, the system analyses the queries in order to get the appropriate ontological representation. Effectively, the system finds the knowledge representations that best match the concepts in order to get the relevant KS from the knowledge repository for a given query.

IV. TECHNICAL ARCHITECTURE & CHOICES

The technical architecture supporting the software infrastructure conceived here as our proof of concept is structured in three main layers: Knowledge Repository, Knowledge Services, and User Interface.

Knowledge repository layer holds the domain knowledge, creating a sort of knowledge space, which is organised around three key entities: Knowledge Sources, their respective Knowledge Representations, and the Ontology itself, which comes with its ambassador, the Ontology Server.

Knowledge Sources are elements which represent the corporate memory of an organization, i.e., documents, spread sheets, media files, and similar sources that can be used to support the acquisition or creation of knowledge. The KS repository represents, then, the collection of all KS currently available in the knowledge space.

When a new KS is added into the knowledge space, its respective knowledge representation is created by the system in order to characterise such KS. The knowledge representation includes some basic information about the KS and adds its specific semantic vector. Broadly speaking, a semantic vector (which will be described in detailed in further sections) gives the best ontological representation to index the KS just added into the space. Therefore, the knowledge representations repository is a container that aggregates all knowledge representations currently available in the knowledge space.

The ontology holds concepts, axioms and relations used to represent knowledge in the domain of work. In our case, the ontology is structured as a pair of taxonomies, as follows: (i) taxonomy of concepts connected via pure taxonomical relations (e.g., as is); and (ii) taxonomy of relations, which contains ontological relations (other than the pure taxonomical ones) also used to improve the semantic links of ontological concepts. These taxonomies are used in different phases of the semantic vector creation, which is also described in detail in further sections. The ontology server is then a software component acting as the ontology ambassador, which means, it provides the way to access to any ontological data.

The knowledge services layer offers the key semantic services used in the knowledge space, namely indexing, discovery, and maintenance, which are respectively provided by the following components: Indexer, Discover, and Maintener. From interoperability point of view, it is worth mentioning that knowledge services are provided as a set of web-services.

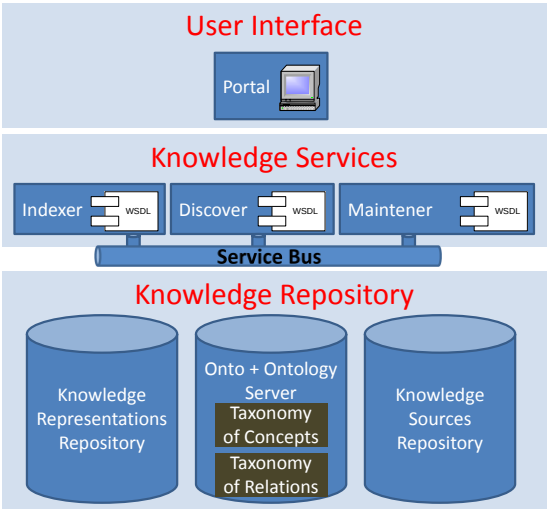


Figure 2. Technical architecture

The User Interface layer offers the front-end with the user via web portal, enabling users to interact with the knowledge space.

In terms of technical choices, two points are highlighted. Firstly, the adoption of the Web services model also plays a very strategic role regarding openness, interoperability, and integration of the system. We use Web Services Description Language (WSDL) [18] to specify the knowledge services, which can also be used to integrate any additional service deemed necessary to our system and which can be provided by third party. Having the WSDL file describing a given web service it is easy to produce the web client able to invoke that service. Thanks to this mechanism, all knowledge services currently provided are available to any web application in the same way that the system interoperates with any other web application.

Secondly, the Java language was chosen due to its key features, which are platform independence and open source model.

A. The Ontology

Knowledge sources strongly rely on ontological concepts, as a way to reinforce their semantic links. The ontology uses a taxonomy of concepts holding two dimensions: on one hand, the knowledge sources themselves are represented in a tree of concepts and, on the other hand, the industrial domain being considered. Instances of concepts (also called individuals) are used to extend the semantic range of a given concept. For instance, the ontological concept of 'Design_Actor' has two instances to represent architect and engineer as roles that can be considered when dealing with knowledge sources (see Figure 3) related to design (experts, design-related issues/solutions, etc.). Moreover, each ontological concept also includes a list of terms and expressions, called equivalent terms, which may represent synonyms or expressions that can lead to that concept. Ontology support is particularly useful in terms of indexation and classification towards future search, share and reuse.

Figure 3. Instances of Knowledge Sources.

The ontology is developed to support and manage the use of expressions which contextualize a KS within the knowledge repository. The ontology adds a semantic weight to relations among KS stored into the knowledge repository. Every ontological concept has a list of 'equivalent terms' that can be used to semantically represent such concept. These terms are, then, treated in both statistical and semantic way to create the semantic vector that properly indexes a given KS.

The ontology was not developed from scratch; rather, it has been developed taking into account relevant sources of inspiration, such as the buildingsmart IFD model [10], [11], and the e-cognos project [12].

The basic ontological definition is as follows: a group of Actors uses a set of Resources to produce a set of Products following certain Processes within a work environment (Related Domains) and according to certain conditions (Technical Topics). Other domains define all relevant process attributes. For example, the Technical Topics domain defines the concepts of productivity, quality standard and duration.

B. The Services

The semantic support services that compose the API layer can globally be described as the following:

- Indexing: The service is designed to accept a list of keywords, compare the keywords to ontological concepts, and produce a ranked list of ontological concepts that best matches that list of keywords. For each keyword, it calculates a corresponding weight reflecting its relevance. The set of keyword-weight pairs is the semantic vector of the knowledge source. This vector is then used to assign a hierarchy of relevant metadata to each knowledge source.
- Discover: The service enables the user to perform searches across knowledge elements, is invoked whenever a user requests a search for a set of keywords. The service produces a matching ontological concept for these keywords, and then matches the resulting concept to the metadata of target knowledge source. This ontology-centred search is the essence of semantic systems, where search phrases and semantic vectors are matched through ontological concepts.
- Maintener: This service is responsible for managing the domain ontology enabling the following capabilities: Browse the concepts/relations(allows navigation through the ontology, showing the description of both concepts and relations); Create new concept(allows the addition of a new concept into the ontology); Create new relation(allows the addition of a new relation into the ontology); Create new attribute(allows the addition of a new attribute to a concept); Import OWL ontology; and Remove concept(allows removal of a concept from the ontology).

 ISBN: 978-1-61208-175-5

V. INDEXATION PROCESS

To better understand the indexation process through semantic vectors comparison (Figure 4), it is necessary to understand how and where these are created and used.

Each semantic vector contains the necessary ontological concepts that best represent a given knowledge source when it is stored into the knowledge repository. These concepts are ordered by their semantic relevance regarding the KS. KS are compared and matched based on their semantic vectors and the degree of resemblance between semantic vectors directly represents the similarity between KS.

Semantic vectors are automatically created using project-related knowledge, using a process which collects words and expressions, to be matched against the equivalent terms which represent the ontological concepts. This produces an inventory of: (i) the number of equivalent terms matched at each ontological concept; and (ii) the total number of equivalent terms necessary to represent the harvested knowledge. This inventory provides the statistical percentage of equivalent terms belonging to each ontological concept represented in the universe of harvested knowledge. This step represents, the calculus of the 'absolute' semantic vector of a given KS, taking into account the equivalent terms-based percentages.

However, the approach presented here also considers a configurable hierarchy of KS relevance, as part of the creation of semantic vectors. This hierarchy is defined using 'relative' semantic factors to all types of KS, which ranges respectively from low relevance (0) to high relevance (1) for the context creation. Both hierarchy and relative semantic factors can be changed if necessary, depending on what KS are considered most relevant for the indexation process.

The final step, which comprehends the semantic evaluation, also includes ontological concepts that are not linked to the knowledge gathered, but have a semantic relationship of proximity with a relevant (heavy) ontological concept. This is done through the definition of a secondary semantic factor to ontological concepts based on their relative distances, inside the ontology tree.

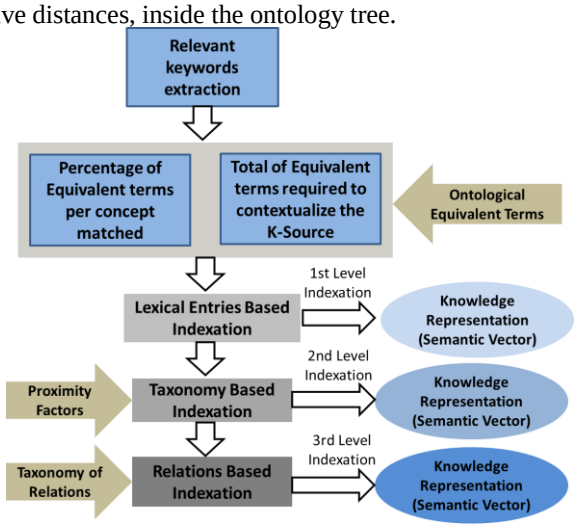


Figure 4. Semantic Vector creation process

Summing up, the final calculation of the semantic vector includes: statistical percentages based on the equivalent terms, the hierarchy of relevance for KS, and the weight assigned to the proximity level.

As referred previously, semantic vectors are continuously updated through the project's life cycle, and even in project's post-mortem. This is done in order to maintain the semantic vector's coherence with the level of knowledge available. Semantic vectors are automatically created: (i) whenever a new KS is gathered; and (ii) to help answering queries issued by the users.

Our approach provides three algorithms to perform the process of retrieving the best ontological representation and weight for both the KS and the query. Those algorithms, namely "Lexical Entries based", "Taxonomy based" and "Relation based" work as follows:

- Lexical Entries: each concept is defined with a list of lexical entries in a different language. The algorithm gets all the lexical entries of all the concepts of the ontology and matches them with all the keywords in the semantic vector. Therefore at the end of the first step, a list of concepts (Lc) matching the semantic vector is built. Further, the weight of the concept C (Wc) is calculated for all the concepts in the list applying the following formula:

$$Wc = NKm \div NKsm \quad (1)$$

where:
Wc: weight calculated to the concept.
NKm: number of keywords that match the concept C.
NKsm: number of keywords in the semantic vector.

- Taxonomy: The algorithm starts from the list "Lc" built in the "Lexical Entries based" algorithm and provides a different way to arrive at the weights. The aim is to try to increase the weight of the concepts which may have received a poor weight in the first stage trying to see if they are close in the taxonomy to a concept that received a good weight in the first stage. The "Lc" list gets the best concepts that match the keywords. A concept is considered a best concept when its weight exceeds the value "best-concept-range" defined in the parameters table. The others are named "worth concepts". For each best concept, the algorithm checks if there are worth concepts nearby concepts of the Lc list in the taxonomy. If this is the case, their respective weights are augmented according the following formula:

$$Wc = Wbc \times Vp \quad (2)$$

where:
Wc: weight performed for the concept
Wbc : weight of the best concept
Vp: value got in the parameters table depending on the level and the way.

The Vp is a value between 0 and 1 and depends on the distance between the best concept and the worth concept in

 ISBN: 978-1-61208-175-5

the hierarchy of concepts. The weight of the new concept is only updated in case the weight given to the preformed concept is greater than the old one. This step is implemented as a way to promote concepts that are strongly taxonomically-related with the best matched concept. Analogously, other concepts that are not so strongly taxonomically-related with the best concept match are penalized.

- Relation: this algorithm will be available in the next version of the system. It aims to integrate the richness of the relations among the concepts in order to provide a more powerful way to represent a KS.

A. *Example*

The list of ontological concepts that best represents a knowledge source is ranked according to the ontological weights assigned to each concept. As stated, there are three ways to calculate such a weight, namely: equivalent terms-based, taxonomy-based, and fully ontology-based. The equivalent terms represent the keywords related to each concept (synonyms or words that can be associated to that concept). They are then used as "indexes" to access the concepts, therefore using purely "statistics" (the greater the number of equivalent terms of a given concept found in the KS representation, the heavier the concept becomes). The taxonomy-based way takes the previous weight and refines it using the "is a" relation to navigate around the heaviest concepts and augment the weight of neighbouring concepts (this augmentation is based on a configurable table of factors guiding generalization/specialization of the taxonomy). The fully ontology-based method exploits all the relations that start from the heaviest concepts to augment the neighbouring concepts (augmentation process is similar to the taxonomy-based one).

Figure 5 and Figure 6, present the results of calculating weights using the equivalent term- and the taxonomy-based methods. The KS representation is given by such concepts: "Heat Pump; Product; Cooling Tower; Solar Collector; Climate Control; Central Heat Generator; Waste Management; Transformation and Conversion; Fan; Extractor; Air Ductwork; Steam Treatment" (column Keywords) and the respective concepts found that match it (column Concepts). The column “Lexical” show the first weight calculated, that is the ratio between number of keywords related to one concept and the total number of keywords in the query (e.g., for the first concept, Transformation and Conversion, the value 0.417 comes from 5 divided by 12). This is a very straight forward calculation.

TABLE I. LEXICAL TERMS VS TAXONOMY BASED

Concepts(7)	Keywords(12)	Keywords(#)	Lexical	Taxonomy
Transformation and Conversion	Heat Pump; Cooling Tower; Solar Collector; Central Heat Generator; Transformation and Conversion	5	0,417	0,417
Product	Product	1	0,083	0,055
Climate Control	Climate Control	1	0,083	0,25
Waste Management	Waste Management	1	0,083	0,055
Impelling Equipment	Fan, Extractor	2	0,167	0,111
HVAC Distribution Device	Air Ductwork	1	0,083	0,055
Energy Treatment	Steam Treatment	1	0,083	0,055

Figure 5 shows the comparative results of the two methods. It is possible to detect immediately that some concepts have had their weights increased. After calculating the first weight, the taxonomy-based method is applied, where it is evaluated the neighbourhood of the heaviest concept(s) and, by following their taxonomical relations, raises the weight of the neighbouring concepts.

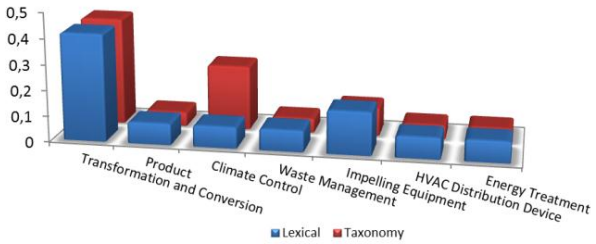


Figure 5. Comparison between equivalent term- and taxonomy-based weights

The initialization of such weights is done manually using a table of values that expresses the factors to be used when augmenting a super/sub concept. This table is configured by the user. It is worth noticing that the taxonomy-based method always keeps the higher weight if the taxonomy-based weight is going to be smaller than the equivalent term-based weight.

Finally, Figure 6 illustrates how the taxonomical relations are used to raise the weight of neighbouring concepts.

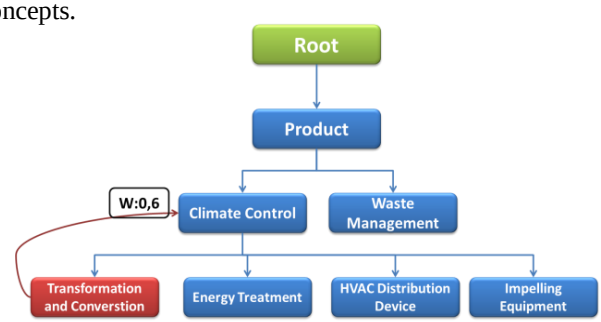


Figure 6. Using the taxonomy to calculate ontological weights.

 ISBN: 978-1-61208-175-5

In this example "Transformation and Conversion" is the heaviest concept with a weight of 0.417, having the neighbour "Climate Control". Therefore, using the factor configured by the user (0.6) the weights of the neighbours are recalculated. As consequence, "Climate Control" is augmented whilst "Product", "Waste Management", "Impelling Equipment", "HVAC Distribution Device", "Energy Treatment" are penalized proportionally.

VI. RESULTS AND ACHIEVEMENTS

This research work is still an on-going process, where relevant empirical data and conclusions aren't not yet matured. An assessment which compares our solution with already existing ones is not yet developed, due to the fact the empirical data and conclusions can't be drawn yet. Work developed so far includes the establishment of the lexical entries and taxonomy based algorithms and the improvement of the taxonomy based algorithm with the inclusion of heuristics which enable the weights associated with the taxonomy relations to change dynamically.

VII. CONCLUSIONS

This work brings a contribution focused on collaborative engineering projects where knowledge engineering plays the central role in the decision making process.

Key focus of the paper is the indexation and retrieval of knowledge sources provided by semantic services enabled by a domain ontology. This work specifically addresses collaborative engineering projects from the Construction industry, adopting a conceptual approach supported by knowledge-based services. The knowledge sources indexation process is supported using a semantic vector holding a classification based on ontological concepts.

When addressing collaborative working environments, there is a need to adopt a semantic description of the preferences of the users and the relevant knowledge elements (tasks, documents, roles, etc.). In this context, we foresee that knowledge sources can be semantically enriched when adopting the indexation process described within this work

Ontologies which support semantic compatibility for specific domains should be adaptive and evolving within a particular context. Ontologies ability to adapt to different environments and different context of collaboration is of extremely importance, when addressing collaborative engineering projects at the organizational level.

As future work regarding this research topic, there is a need to further analyse into what extent neighbours concepts can influence the calculus of the semantic vector as well as how ontological relations can contribute also to the better representations of knowledge given by the semantic vector.

REFERENCES

[1] I. Nonaka and H. Takeuchi, The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. New York: Oxford University Press, 1995

[2] J. Firestone and M. McElroy, Key Issues in the New Knowledge Management. Burlington: Butterworth-Heinemann, 2003.

[3] M. Koenig, The third stage of KM emerges. KMWorld, March 2002.

[4] Y. Malhotra and Y, Beyond "Hi-Tech Hidebound" Knowledge Management: Strategic Information Systems for the New World of Business. BRINT Research Institute, 1999.

[5] M. McElroy, "The Second Generation of Knowledge Management," Knowledge Management Magazine, Oct 1999, pp. 86-88.

[6] K. Dalkir, Knowledge Management in Theory and Practice. Oxford: Elsevier, 2005.

[7] A. Zeeshan, A. Chimay, R. Darshan, P. Carrillo, and D. Bouchlaghem, "Semantic web based services for intelligent mobile construction collaboration," ITcon, 367-379, 2004.

[8] O. Lassila and M. Adler, "Semantic Gadgets: Device and information interoperability". Cleveland: Ubiquitous Computing Environment workshop, 2003.

[9] R. Costa, C. Lima, J. Antunes, P. Figueiras, and V. Parada, "Knowledge management capabilities supporting collaborative working environments in a project oriented context," European Conference on Intellectual Capital (pp. 208-216). Lisbon: ACI, 2010.

[10] buildingSMART. Retrieved: September, 2011 from IFD Library: http://www.ifd-library.org/

[11] OCCS. Retrieved: September, 2011 from OmniClass: http://www.omniclass.org/

[12] C. Lima, B. Fies, T. Diraby, and G. Lefrancois, "The challenge of using a domain Ontology in KM solutions: the e-COGNOS experience," International Conference on Concurrent Engineering: Research and Applications, (pp. 771-77). Funchal, 2003.

[13] P. Castells, M. Fernandez, and D. Vallet. An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. IEEE Trans. on Knowl. and Data Eng. 19, 2 (February 2007), pp. 261-272.

[14] R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval, Addison Wesley, 1999.

[15] V. Gudivada, V. Raghavan, W. Grosky, and R. Kasanagottu. Information retrieval on the World Wide Web. Internet Computing, IEEE , vol. 1, no.5, pp. 58-68, Sep/Oct 1997

[16] F. Burkowski, "An algebra for hierarchically organized text-dominated databases", Information Processing & Management, 28(3), 333–348, 1992

[17] G. Navarro and R. Baeza-Yates. Proximal Nodes: A Model to query document databases by contents and structure, ACM Transactions on Information Systems 15 (4), October 1997, pp. 401-435.

[18] Web Services Description Language (WSDL) Retrieved: September, 2011 from World Wide Web Consortium: http://www.w3.org/TR/wsdl

 ISBN: 978-1-61208-175-5

A Fuzzy Logic Semantic Mapping Approach for Fuzzy Geospatial Ontologies

Mohamed Bakillah
Geomatics Research Center
Laval University
Québec, Canada
mohamed.bakillah.1@ulaval.ca

Mir Abolfazl Mostafavi
Geomatics Research Center
Laval University
Québec, Canada
Mir-abolfazl.mostafavi@scg.ulaval.ca

***Abstract*—The problem of finding semantic mappings between heterogeneous geospatial databases is a key issue in the development of a semantic interoperability approach. An essential step towards the success of a semantic approach is the ability to take into account the fuzzy nature of geospatial concepts being compared and of the semantic mapping process itself. While fuzzy ontologies and quantitative fuzzy matching methods have been proposed, they are not targeted at the geospatial domain. In this paper, we present a fuzzy semantic mapping approach for fuzzy geospatial ontologies, which employs fuzzy logics. The fuzzy semantic mapping approach has the capability to produce fuzzy qualitative semantic relations between concepts of fuzzy ontologies, which are richer than quantitative-only matches that are provided by existing approaches. In an application example, we show how fuzzy mappings can be used to propagate fuzzy queries to relevant sources of a network. In this way, the fuzzy semantic mapping supports geospatial data sharing among remote databases of the network while taking into account uncertainties that are inherent to the geospatial concepts and the semantic interoperability process.**

Keywords-semantic interoperability; fuzzy logics; fuzzy geospatial ontology; semantic mapping

I. INTRODUCTION

The spreading of decentralized systems has created the need for approaches supporting users to find the relevant sources that can provide the data they required. Furthermore, an important number of users search for geospatial data, e.g. "flooding risk zones near built-up areas of Montreal." Geospatial ontologies are considered as useful tools to support the identification of relevant geospatial data sources [1][2][3][4]. For example, Cruz et al. [5] indicate that the problem of querying geospatial databases in a distributed environment can be addressed by finding semantic mappings between the ontologies that describe each database.

However, several recent researches in GIScience have acknowledged the need for representing and dealing with the uncertainty and fuzziness of geospatial phenomena [6][7][8][9][10]. For example, a flooding risk zone is a fuzzy concept because different sources can define it with different characteristics.

Consequently, geospatial ontologies have to support the representation, but until now, the representation of fuzziness in ontologies has been mostly limited to the non-geospatial domain [11][12][13]. In addition, in order to resolve semantic heterogeneity among fuzzy geospatial ontologies, there is a need for a semantic mapping approach that will be able to deal with fuzzy geospatial ontologies.

We propose that fuzzy logic is well adapted for representing fuzzy knowledge about geospatial concepts, provided that the representation of concepts is explicit enough and takes into account all spatiotemporal aspects of concepts. In this paper, we propose a solution to the problem of fuzzy geospatial ontology and fuzzy semantic mapping. We first provide a definition of what is a fuzzy geospatial ontology. Then, we propose a new fuzzy semantic mapping approach, which takes as input the concepts of the fuzzy geospatial ontologies and finds semantic relations between concepts and their degree of fuzziness. The fuzzy semantic mapping approach integrates fuzzy logic operators and predicates to reason with fuzzy concepts. Finally, we demonstrate a possible application of the fuzzy semantic mapping, which is the propagation of fuzzy queries to the relevant sources of a network.

This paper is organized as follows. In Section 2, we discuss the role of fuzzy theory in semantic interoperability for GIS. In Section 3, we present the definition of the fuzzy geospatial ontology. In Section 4, we propose the fuzzy semantic mapping approach. In Section 5, we present the application for query propagation. In Section 6, we conclude this paper.

II. ROLE OF FUZZY THEORY IN SEMANTIC INTEROPERABILITY OF GEOSPATIAL DATA

Semantic interoperability is a major research topic to ensure data sharing among different geospatial databases in a network [14][15]. Semantic interoperability is the knowledge-level interoperability that provides cooperating databases with the ability to resolve semantic heterogeneities arising from differences in meanings of concepts [16]. Semantics, which is the meaning of expressions in a language [17] [18], is crucial for semantic interoperability because two systems can "understand" each other and share knowledge only if they make the meaning of their concepts apparent to each other. Ontologies, which are explicit specifications of a conceptualisation [19], aim at capturing semantics of data [20] [21][22] [14][23] [24]. Ontologies with poor (implicit) semantics provide weaker interoperability while ontologies with strong semantics based on logical theory support richer semantic interoperability

 ISBN: 978-1-61208-175-5

[25]. On the other hand, uncertainty in the semantics of concepts should be considered as a kind of knowledge that must also be explicit in conceptual representations, as argued by Couclecis [7]. Fuzzy logic proposed by Zadeh is considered in GIScience as a suitable way to represent uncertain knowledge and reason with it. Therefore, several approaches have proposed to augment ontologies with fuzziness, for example for news summarization [11], for information retrieval in the medical domain [12], or for image interpretation [13]. However, these approaches are not targeted at the geospatial domain. For example, geospatial concepts are often described with properties (e.g., "inclination" of "lowland"), which range of values can be fuzzy. However, existing fuzzy ontology representations and ontology mapping approaches do not consider properties with fuzzy range of values. Other approaches in the geospatial domain use fuzzy sets to assess similarity of categorical maps [26]. But this approach is not general and aims at categorical maps, while we argue that a more general framework for any geospatial fuzzy ontology is needed. In addition, we argue that quantitative fuzzy similarity have limited expressivity in comparison to qualitative semantic relations, which are easier to interpret by users. To our knowledge, there is no existing fuzzy semantic mapping approach that produces fuzzy semantic relations. In our paper, we propose a definition of the fuzzy geospatial ontology, and an approach that addresses this need.

III. FUZZY GEOSPATIAL ONTOLOGY

An ontology is usually defined as a set of concepts (or classes) that represent entities of the domain of discourse, relations and/or properties, and axioms that indicate statements that are true within that domain of discourse [14]. An example of axiom is "all intersections involve at least two roads." We follow a similar approach to define the fuzzy geospatial ontology. However, in the fuzzy ontology, we consider that membership of a property or relation in the definition of a concept can be quantified. In a crisp ontology, the membership degree of a property of relation into the definition of a concept is always one or zero. This means that either a concept has that property; or it does not have it. In the fuzzy ontology, this membership degree varies between zero and one, to indicate partial membership. Therefore, in a fuzzy ontology, concepts do not have a fully determined definition.

We define the fuzzy geospatial ontology as a 5-tuple: $O = \{C, R, P, D, rel, prop\}$, where C is a set of concepts, which are abstractions of entities of the domain of discourse; R is a set of relations; P is a set of properties for concepts; D is a set of possible values for properties in P, called range of properties; rel: $[R \rightarrow C \times C] \rightarrow [0, 1]$ is a fuzzy function that specifies the fuzzy relation that holds between two concepts; prop: $[P \rightarrow C \times D] \rightarrow [0, 1]$ is a fuzzy function that specifies the fuzzy relation between a concept and a subset of D. D is therefore a fuzzy range of values. The set of relations R includes relations such as "has geometry," which indicates the geometry of instances of the concept, such as polygon, moving polygon, line, and other GML spatial and spatiotemporal types. It also includes spatial relations such as "Is_located_at," which indicates the location of an instance of the concept, and other topological, directional and orientation spatial relations. An example of fuzzy property "inclination" of the fuzzy concept "lowland" is given on Fig. 1. Lowlands are regions which inclination is relatively flat, but there is a certain level of fuzziness when we try to determine if a given region is a "lowland." While the value "flat" of the "inclination" property has the fuzzy membership of 0.8 to the range of values of "inclination," the value "low" has a lower membership value of 0.10. This reflects the fact that a greater percentage of lands with flat inclination are considered as members of the geographical category "lowland,' in comparison to lands with "low" inclination.

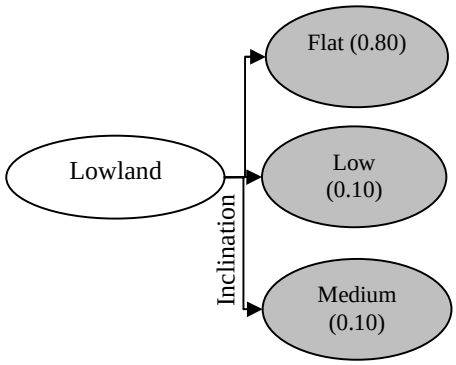


Figure 1. Example of fuzzy property "inclination" for concept "lowland"

For the purpose of our approach, we define a concept with a conjunction of a set of axioms A_C, where each axiom is a fuzzy relation or property that defines the concept:

$$C = A_1 \sqcap A_2 \sqcap \ldots. \sqcap A_n.$$

We use the term axiom, which is usually employed to refer to the whole expression that defines a concept, because a concept could also be defined by one feature (property or relation).

IV. FUZZY SEMANTIC MAPPING PROCESS

In this section, we propose the new fuzzy semantic mapping approach. The idea of this approach is to use fuzzy logics to first determine the fuzzy inclusion of a concept into another concept from a different ontology, based on the fuzzy inclusion of each axiom of the first concept into axioms of the second concept. Then, fuzzy predicates, which value depends on the fuzzy inclusion, are used to infer the semantic relation between the two concepts.

Let two concepts C and C', defined as follows:

$$C = A_1 \sqcap A_2 \sqcap \ldots. \sqcap A_n$$

$$C' = A_1' \sqcap A_2' \sqcap \ldots. \sqcap A_m'$$

 ISBN: 978-1-61208-175-5

We define the fuzzy semantic mapping between C and C' as follows:

Definition (fuzzy semantic mapping) A fuzzy semantic mapping m^C between *C* and *C*' is a tuple $m^C = <C, C', rel(C, C'), \mu(C, C')>$, where *rel* is a semantic relation between *C* and *C*', and $\mu(C, C')$ is the fuzzy inclusion of *C* into *C*'.

First, we explain how the fuzzy inclusion of *C* into *C*' is computed. Secondly, we explain how the semantic relation *rel* between *C* and *C*' is determined.

A. Fuzzy inclusion

We define the fuzzy inclusion as the membership degree of a concept in another. This means that when the value of the fuzzy inclusion is 1, the first concept is entirely included in the second concept; when it is zero, no axiom of the first concept intersects with axioms of the second. The fuzzy inclusion of C into C' is denoted with $\mu(C, C')$:

$$\mu(C,C') = \frac{\sum_{A\in(A_1,...,A_n,A_1',...A_m')} \min(\mu_C(A), \mu_{C'}(A))}{\sum_{A\in(A_1,...,A_n,A_1',...A_m')} \mu_C(A)}, \quad (1)$$

where $\mu_C(A)$ is the membership degree of axiom A in concept C. We know that this membership degree comes from the definition of the concept in the fuzzy geospatial ontology. Let A: <r.D> and A': <r'.D'> be two axioms, where D and D' are fuzzy domains. For example, <shape.((0.2, circle);(0.8, ellipse))> represents the fuzzy relation on Fig. 1.

To compute (1), which relies on the membership of axiom A in concept C', and where axiom A of concept C might not be already in the definition of the concept C', we need the membership of axiom A in axiom A' of C'. The membership degree of A into A' is determined by the Zadeh conjunction for fuzzy sets:

$$\mu(A,A') = \min(\mu(D,D'), \mu(r,r')). \quad (2)$$

The function $\mu(X1, X2)$ over any fuzzy sets X1, X2 is defined as follows, using the fuzzy implication principle of fuzzy logics [27]:

$$\mu(X1,X2) = \inf_{x\in X1\cup X2}(\mu_{X1}(x) \Rightarrow_f \mu_{X2}(x)), \quad (3)$$

where $\Rightarrow_f$ is a fuzzy implication operator from [0,1] into [0,1]. There are several definitions for the fuzzy implication operator (including Gödel, Gogen and Lukasiewicz fuzzy implications, see Bosc and Pivert [27]). We use Lukasiewicz fuzzy implication because of its superior flexibility, which is defined as follow:

$$\mu_{X1}(x) \Rightarrow_L \mu_{X2}(x) = \begin{cases} 1 & if\ \mu_{X1}(x) \le \mu_{X2}(x). \\ 1-\mu_{X1}(x)+\mu_{X2}(x) & otherwise \end{cases} \quad (4)$$

To compute $\mu(D, D')$ with (3), we use the Lukasiewicz fuzzy composition operator, denoted with the symbol $\otimes$, and which determines the membership of a first element ε_i' in a set D, knowing the membership degree of ε_i' in ε_j and the membership degree of ε_j in D (Fig. 2). The symbol ε is used to indicate an element of the range of values of a property or a relation of the fuzzy geospatial ontology.

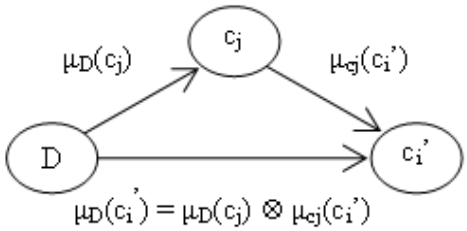


Figure 2. Fuzzy composition principle

The membership degree of ε_i' in D writes as:

$$\mu_D(\varepsilon_i') = \sum_j \mu_D(\varepsilon_j) \otimes \mu_{\varepsilon_j}(\varepsilon_i'),\ \forall j \left| (\neg\varepsilon_j \perp \varepsilon_i), \right. \quad (5)$$

where

$$\mu_D(c_j) \otimes_L \mu_{c_j}(c_i') = \max(\mu_D(c_j) + \mu_{c_j}(c_i') - 1, 0), \quad (6)$$

according to Lukasiewicz's definition of the fuzzy composition operator.

To determine $\mu_{\varepsilon j}(\varepsilon_i')$, which is the membership degree of an element ε_i' of a range of values in an element ε_j of another range of values, we have developed a fuzzy membership degree measure. This measure is based on the relative position of ε_j and ε_i' in an upper-level ontology O. An appropriated ontology for this task would be a domain-independent, largely recognized lexical base, such as WordNet. However, other more specialized upper-level ontologies might be more useful, depending on the domain of application. Let $<_O$ be a hierarchical, is-a relationship between terms in O, such that $t <_O t'$ means that t is more specific (less general) than t'. Let $P(\varepsilon_j, \varepsilon_i')$ be the path relating ε_j to ε_i' in O, according to this hierarchy: $P(\varepsilon_j, \varepsilon_i') = \{\varepsilon_j, t1, t2, \ldots \varepsilon_i'\}$ so that t1, t2, … is the ordered set of nodes from ε_j to ε_i' in O. Let $d(t_k)$ the set of descendants of a node in O. We define $\mu_{\varepsilon j}(\varepsilon_i')$ as follows:

$$\mu_{\varepsilon_j}(\varepsilon_i') = \begin{cases} 1 & \text{if } \varepsilon_i \prec \varepsilon_j \\ \dfrac{1}{\prod_{\forall t_k \in P(\varepsilon_j, \varepsilon_i')} |d(t_k)|} & \text{if } \varepsilon_i' > \varepsilon_j \\ 0 & else \end{cases} . \quad (7)$$

This equation means that, when ε_i' is more specific than ε_j, it is entirely included in ε_j, and when ε_i' is more general than ε_j, $\mu_{\varepsilon j}(\varepsilon_i')$ decreases with the number of descendants of its subsumers. Replacing results of (7) in (6), we obtain the membership of each element of the fuzzy range D' in D, which, in turn, allows to determine μ(D, D') with (3). Eq. (7) is also used to determine μ(r, r'), so these results can be replaced in (3).

From the fuzzy inclusion given in (2), we obtain the semantic relation between the axioms, rel(A, A'), using the following rules, which are derived from the fuzzy set relationship definitions:

(R1) A $\equiv$ A' $\Leftrightarrow$ μ(A, A') =1 $\wedge$ μ(A', A) =1

(R2) A $\sqsubseteq$ A' $\Leftrightarrow$ μ(A, A') =1 $\wedge$ μ(A', A) <1

(R3) A $\sqsupseteq$ A' $\Leftrightarrow$ μ(A, A') <1 $\wedge$ μ(A', A) =1

(R4) A $\sqcap$ A' $\Leftrightarrow$ 0 < μ(A, A') <1 $\wedge$ 0 < μ(A', A) <1

(R5) A $\perp$ A' $\Leftrightarrow$ μ(A, A') =0 $\wedge$ μ(A', A) =0.

B. Semantic relation

In order to determine the semantic relation between concepts, we have defined a set of predicates. The semantic relation between two concepts is determined by the following expression:

$$rel(C, C') = I(A_C, A_{C'}) \otimes_{Pr} C(A_C, A_{C'}) \otimes_{Pr} CI(A_C, A_{C'}),$$

where $I(A_C, A_C')$, $C(A_C, A_C')$ and $CI(A_C, A_C')$ are three predicates that respectively evaluate the intersection of axioms of the concept *C* with axioms of *C'*, the inclusion of axioms of *C'* in axioms of *C*, and the inclusion of axioms of *C* in axioms of *C'*. We have defined a composition operator, denoted $\otimes_{Pr}$, the function of which is to give the semantic relation between *C* and *C*', based on the value of those three predicates. The composition operator takes as input the value for the three predicates for *C* and *C*', and returns the semantic relation between *C* and *C*'.

For any predicate *Pr*, the possible values of *Pr* are:

- *B* value, if for all axioms of *C* there is an axiom of *C'* that verifies predicate *Pr*, and vice-versa. For example, $I(A_C, A_C') = B$ if for all axioms in A_C, there is an axiom in A_C' that intersects this axiom (as determined by rules R1 to R5 defined in the previous section), and vice-versa;
- *S* value, if there exist some axioms of *C* and axioms of *C'* that verify predicate *Pr*, but not all;
- *N* value, if there exists no axiom of *C* and *C'* that verifies predicate *Pr*.

These principles for determining the value of a predicate are formalized as follows (where logic symbols are ∀(for all), ∃ (there exists) ⊥ (disjoint) and ¬ (negation):

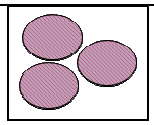
$$I(C,C') = \begin{cases} B & \forall i \exists j, rel(A_i, A'_j) \neq \perp \wedge \mu(A_i, A'_j) \neq 0) \wedge \\ & \exists i \forall j, rel(A_i, A'_j) \neq \perp \wedge \mu(A_i, A'_j) \neq 0) \\ S & \exists i \exists j, rel(A_i, A'_j) \neq \perp \wedge \mu(A_i, A'_j) \neq 0) \wedge \\ & \neg[\forall i \exists j, rel(A_i, A'_j) \neq \perp \wedge \mu(A_i, A'_j) \neq 0) \wedge \\ & \exists i \forall j, rel(A_i, A'_j) \neq \perp \wedge \mu(A_i, A'_j) \neq 0)] \\ N & \neg \exists i \exists j, rel(A_i, A'_j) \neq \perp \wedge \mu(A_i, A'_j) \neq 0) \end{cases}$$

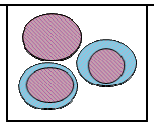
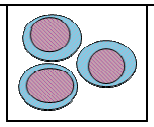
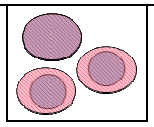
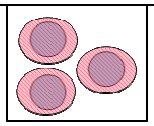
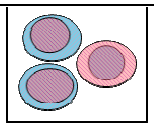
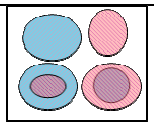
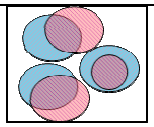
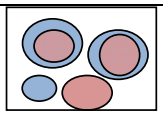
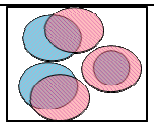
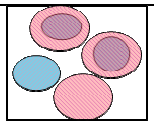
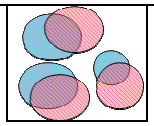
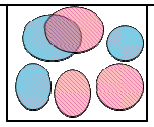
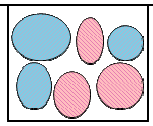
$$C(C,C') = \begin{cases} B & \forall i \exists j, rel(A_i, A_j') \in \{\equiv, \supseteq\} \wedge \mu(A_i, A_j') \neq 0 \\ S & \exists i \exists j, rel(A_i, A_j') \in \{\equiv, \supseteq\} \wedge \mu(A_i, A_j') \neq 0 \wedge \\ & \neg \forall i \exists j, rel(A_i, A_j') \in \{\equiv, \supseteq\} \wedge \mu(A_i, A_j') \neq 0 \\ N & \neg \exists i \exists j, rel(A_i, A_j') \in \{\equiv, \supseteq\} \wedge \mu(A_i, A_j') \neq 0 \end{cases}$$

$$CI(C,C') = \begin{cases} B & \forall i \exists j, rel(A_i, A_j') \in \{\equiv, \subseteq\} \wedge \mu(A_i, A_j') \neq 0 \\ S & \exists i \exists j, rel(A_i, A_j') \in \{\equiv, \subseteq\} \wedge \mu(A_i, A_j') \neq 0 \wedge \\ & \neg \forall i \exists j, rel(A_i, A_j') \in \{\equiv, \subseteq\} \wedge \mu(A_i, A_j') \neq 0 \\ N & \neg \exists i \exists j, rel(A_i, A_j') \in \{\equiv, \subseteq\} \wedge \mu(A_i, A_j') \neq 0 \end{cases}$$

For *C* and *C'*, the domain of quantifiers is respectively $i \in \{1,\ldots, n\}$ and $j \in \{1,\ldots, m\}$. When the three predicates are evaluated within {B, S, N}, the result is 14 classes of cases, which are provided in Table 1. This table defines the $\otimes_{Pr}$ operator: each combination of values for the three predicates is associated with a resulting semantic relation. For example, *C* (semantically) contains *C'* if $I(A_C, A_C') = B$, $C(A_C, A_C') = B$ and $CI(A_C, A_C') = S$ (second line of Table 1). In the associated illustrations, blue sets represent axioms of *C*, and red sets axioms of *C'*.

TABLE I. SEMANTIC RELATIONS IN FUNCTION OF THE COMBINATION OF PREDICATE VALUES ($\otimes_{Pr}$ OPERATOR)

Semantic relationship	Value of $I(A_C, A_{C'})$	Value of $C(A_C, A_{C'})$	Value of $CI(A_C, A_{C'})$	Representation
1. Equivalence	B	B	B	

2. Contains	B	B	S	
	B	B	N	
3. Contained In	B	S	B	
	B	N	B	
4. Partial S-Containment (S=Symetric)	B	S	S	
	S	S	S	
5. Partial L-Containment (L-LEFT)	B	S	N	
	S	S	N	
6. Partial R-containment (R=RIGHT)	B	N	S	
	S	N	S	
7. Strong Overlap	B	N	N	
8. Weak Overlap	S	N	N	
9. Disjoint	N	N	N	

Once the semantic relations and fuzzy inclusion are determined between concepts, we aim to show that this information can be used to find relevant sources of a network through propagation of query.

V. APPLICATION EXAMPLE

The presented application aims to demonstrate the usefulness of the proposed approach. As an example of fuzzy ontologies, we consider the ontology fragments in Fig. 3 and Fig. 4.

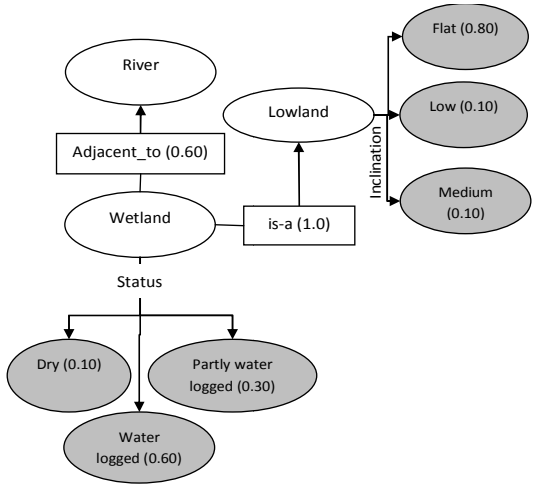


Figure 3. Portions of ontology A for the application example

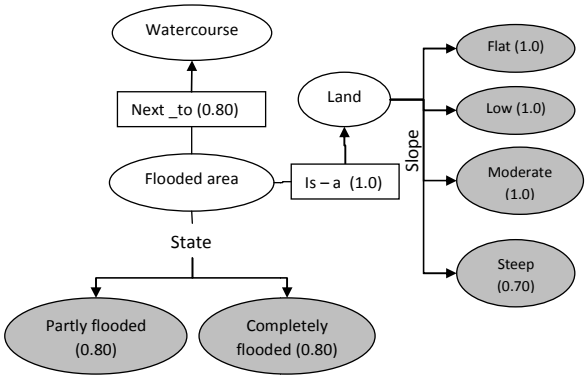


Figure 4. Portions of ontology B for the application example

The fuzzy ontology A describes the concept “wetland” as “lowland” which can have flat, low or medium slope. The values “flat,” “low” and “medium” constitute the range of the property “has slope.” For each of these values, there

 ISBN: 978-1-61208-175-5

is a fuzzy membership value that indicates the degree of membership of the value into the range of the property. Similarly, the wetland has the property "status," which range is composed of values "dry," "partly waterlogged" and "waterlogged."

The fuzzy ontology B describes the concept "flooded area," which has the property "state" with values "partly flooded" and "completely flooded."

Consider that the user of ontology A needs to find data on wetlands in a given region. To do so, the ontological description of concept "wetland" is compared to the ontological description of concepts from other available sources, for instance fuzzy ontology B. The fuzzy matching approach is used for this purpose. The following shows the values that are obtained for the membership of axioms of both ontologies into the concepts of "flooded area" and concept of "wetland:"

$\mu_{flooded\_area}(<is\_a.lowland>) = 0.50$

$\mu_{wetland}(<is\_a.lowland>) = 1.00$

$\mu_{flooded\_area}(<adjacent\_to.river>) = 0.30$
$\mu_{wetland}(<adjacent\_to.river>) = 0.60$

$\mu_{flooded\_area}(<status.dry>) = 0.00$
$\mu_{wetland}(<status.dry>) = 0.10$

$\mu_{flooded\_area}(<status.partly\_waterlogged>) = 0.10$
$\mu_{wetland}(<status.partly\_waterlogged >) = 0.30$

$\mu_{flooded\_area}(<status.waterlogged>) = 0.20$
$\mu_{wetland}(<status.waterlogged >) = 0.60$

$\mu_{flooded\_area}(<is\_a.land>) = 1.00$
$\mu_{wetland}(<is\_a.land>) = 0.50$

$\mu_{flooded\_area}(<next\_to.watercourse>) = 0.80$
$\mu_{wetland}(<next\_to.watercourse>) = 0.30$

$\mu_{flooded\_area}(<state.partly\_flooded>) = 0.80$
$\mu_{wetland}(<state.partly\_flooded>) = 0.10$

$\mu_{flooded\_area}(<state.completely\_flooded>) = 0.80$
$\mu_{wetland}(<state.completely\_flooded >) = 0.20$

When those values are inserted in (1), we obtain that μ(flooded_area, wetland) = 0.69.

The semantic relation between "wetland" and "flooded area" is obtained by computing the three predicates, which values are the following:

$I(A_{flooded\_area}, A_{wetland}) = B$
$C(A_{flooded\_area}, A_{wetland}) = S$
$CI(A_{flooded\_area}, A_{wetland}) = N$

The resulting semantic relation, according to Table 1, is "partial left containment," which means that some axioms in the definition of "wetland" are included in some axioms of "flooded area." The fuzziness of this relation is 0.69.

When the requestor receives a set of concepts that partly matches its query, he or she can select the more relevant concept using two complementary information elements, the semantic relation and the degree of fuzziness of this relation. The fuzziness is more than a semantic similarity, since it takes into account the fuzziness of concepts being compared. For example, a property value which has a low membership degree into the concept's definition, such as "dry" in the above example, will have less "weight" in the computation of the semantic mapping than a property that has higher membership degree, such as "waterlogged."

While the objective of the paper was not to demonstrate the cost of implementing the approach, we note that the concept of fuzzy mapping can be useful to support various semantic interoperability tasks. More particularly, it is an approach that can support query propagation in decentralized environment. In such environment, there is no central authority that can identify the sources that can process a query. Therefore, the goal of query propagation is to forward the query from source to source through an optimal path, i.e. a path that will contain the most relevant sources with respect to the query. The qualitative and quantitative mappings issued by the fuzzy semantic mapping algorithm can be used as criteria to select the sources that are relevant along the path, while taking into account the fuzziness of semantic mappings.

VI. CONCLUSION

In the geospatial domain, it is essential to consider the uncertainty and fuzziness of geospatial phenomena. Establishing semantic mappings between fuzzy geospatial ontologies is still an issue that was not fully addressed. In this paper, we have dealt with some problems related to the representation of fuzziness in geospatial ontologies, and fuzzy semantic mapping between fuzzy geospatial ontologies. In order to address these problems, we have proposed a fuzzy geospatial ontology model, and a new fuzzy semantic mapping approach. The determination of fuzzy semantic mappings is based on fuzzy logics and a set of predicates that were defined to determine fuzzy semantic relations between concepts, which are complementary to the fuzzy inclusion degree between concepts. The qualitative and quantitative results give more information for the user to understand the nature of relation between its fuzzy query and available concepts. One of the possible uses of our approach is query propagation in a network of

 ISBN: 978-1-61208-175-5

heterogeneous, fuzzy geospatial ontologies. Query propagation determines to which nodes of a network a given query should be forwarded in order to obtain optimal query results. Query propagation provides the user with a path in the network that contains the most relevant sources to answer the query. In future work, we will apply this approach to the issue of query propagation. We also plan to extend the fuzzy semantic mapping approach to more complex cases of the fuzzy spatial, temporal and spatiotemporal features of concepts. This is essential for propagating queries to relevant concepts, for if spatiotemporal properties have different meanings, the query may return inaccurate results. In addition, we plan to extend the approach to the case of an ad hoc network, where sources could be added or removed from the network in real time.

ACKNOWLEDGMENT

This research was made possible by an operating grant from Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] R. Lemmens, Semantic Interoperability of Distributed Geo-Services. Ph.D Thesis, International Institute for Geo-Information Science and Earth Observation (ITC), Enschede, The Netherlands, 323 p., 2006.
http://www.ncg.knaw.nl/Publicaties/Geodesy/pdf/63Lemmens.pdf <retrieved: Nov, 2011>

[2] M. Lutz and E. Klien, "Ontology-based Retrieval of Geographic Information," International Journal of Geographical Information Science, vol. 20, 2006, pp. 233–260.

[3] L. Vaccari, P. Schvaiko, and M. Marchese, "A Geo-Service Semantic Integration in Spatial Data Infrastructures," International Journal of Spatial Data Infrastructures Research, vol. 4, 2009, pp. 24-51.

[4] M. Bakillah, M.A. Mostafavi, Y. Bédard, and J. Brodeur, "SIM-NET: A View-Based Semantic Similarity Model for Ad Hoc Networks of Geospatial Databases," Transactions in GIS, 13(5), 2009, pp. 417-447.

[5] I. F. Cruz, W. G. Sunna, and A. Chaudry, "Semi-Automatic Ontology Alignement for Geospatial Data Integration," International Conference on Geographic Information Science (GIScience), LNCS 3234, Springer, 2004, pp. 51-66.

[6] J. Zhang and M. Goodchild, Uncertainty in Geographical Information. London: Taylor & Francis, 2002.

[7] H. Couclelis, "The Certainty of Uncertainty: GIS and the Limits of Geographic Knowledge," Transactions in GIS, vol. 7, issue 2, 2003, pp. 165-175.

[8] V.B. Robinson, "A Perspective on the Fundamentals of Fuzzy Sets and Their Use in Geographical Information Systems," Transactions in GIS, vol. 7, issue 1, 2003, pp. 3-30.

[9] O. Ahlqvist, "Using Uncertain Conceptual Space to Translate Between Land Cover Categories," International Journal of Geographical Information Science, vol. 19, issue 7, 2005, pp. 831-857.

[10] H. Ban and O. Alhqvist, "Representing and Negociating Uncertain Geospatial Concepts – Where Are the Exurban Areas?" Computers, Environment and Urban Systems, issue 33, 2009, pp. 233-246.

[11] C.-S. Lee, Z.-W. Jian, and L.-K. Huang, "A Fuzzy Ontology and Its Application to News Summarization," IEEE Transaction on Systems, Man and Cybernetics, vol. 35, issue 5, 2005, pp. 859-880.

[12] D. Parry, "A Fuzzy Ontology for Medical Document Retrieval," Australasian Workshop on Data Mining and Web Intelligence, 2004, pp. 121-126.

[13] C. Hudelot, J. Atif, and I. Bloch, "Fuzzy Spatial Relation Ontology for Image Interpretation," Fuzzy Sets and Systems, vol. 159, 2008, pp. 1929-1951.

[14] P. Agarwal, "Ontological Considerations in GIScience," International Journal of Geographical Information Science, vol. 19, issue 5, 2005, pp. 501–536.

[15] L. Bian and S. Hu, "Identifying Components for Interoperable Process Models using Concept Lattice and Semantic Reference System," International Journal of Geographical Information Science, vol. 21, issue 9, 2007, pp. 1009–1032.

[16] J. Park and S. Ram, "Information systems interoperability: what lies beneath?" ACM Transactions on Information Systems, vol. 22, issue 4, 2004, pp. 595-632.
http://comminfo.rutgers.edu/~muresan/IR/Docs/Articles/toisPark2004.pdf <retrieved: July, 2011>

[17] W. Kuhn, "Geospatial Semantics: Why, of What, and How?" Journal on Data Semantics III, vol. 3534, 2005, pp. 1–24.

[18] G. R. Fallahi, A. U. Frank, M. S. Mesgari, and A. Rajabifard, "An Ontological Structure for Semantic Interoperability of GIS and Environmental Modeling," International Journal of Applied Earth Observation and Geoinformation, vol. 10, issue 3, pp. 342-357.

[19] T.R. Gruber, "A Translation Approach to Portable Ontology Specification," Stanford, California: Knowledge Systems Laboratory, Technical Report KSL 92-71, 1993.
http://www.ksl.stanford.edu/KSL_Abstracts/KSL-92-71.html.ps <retrieved: Oct, 2011>

[20] J. Brodeur, Y. Bédard, G. Edwards, and B. Moulin, "Revisiting the Concept of Geospatial Data Interoperability within the Scope of Human Communication Process," Transactions in GIS, vol. 7, issue 2, 2003, pp. 243-265.
http://yvanbedard.scg.ulaval.ca/wpcontent/documents/publications/349.pdf <retrieved: Sept, 2011>

[21] W. Kuhn, "Semantic Reference Systems," International Journal of Geographical Information Science, vol. 17, issue 5, 2007, pp. 405–409.

[22] A. Rodriguez and M. Egenhofer, "Determining Semantic Similarity Among Entity Classes from Different Ontologies," IEEE Transactions on Knowledge and Data Engineering, vol. 15, issue 2, 2003, pp. 442–456.
http://www.spatial.maine.edu/~max/across.pdf <retrieved: Nov, 2011>

[23] F. Fonseca, G. Camara, and A.M. Monteiro, "A Framework for Measuring the Interoperability of Geo-Ontologies," Spatial Cognition and Computation, vol. 6, issue 4, 2005, pp. 307-329.
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.92.8084.pdf <retrieved: Oct, 2011>

[24] M. Kavouras and M. Kokla, "Theories of Geographic Concepts," CRC Press, Taylor & Francis Group, 2008.

[25] L. Obrst, "Ontologies for Semantically Interoperable Systems," Proceedings of the 12th international conference on information and knowledge management, New Orleans, LA, USA, 2003, pp. 366-369.
http://semanticommunity.info/@api/deki/files/5953/%3DLeoObrst2003.pdf <retrieved: Oct, 2011>

 ISBN: 978-1-61208-175-5

[26] A. Hagen, “Fuzzy Set Approach to Assessing Similarity of Categorical Maps,” International Journal of Geographical Information Science, vol. 17, issue 3, 2003, pp. 235–249.

[27] P. Bosc and O. Pivert, “About Approximate Inclusion and its Axiomatization,” Fuzzy Sets and Systems, vol. 157, 2006, pp. 1438-1454.

 ISBN: 978-1-61208-175-5

Semantic Annotation Semantically: Using a Shareable Extraction Ontology and a Reasoner

Jan Dědek
Department of Software Engineering
MFF, Charles University
Prague, Czech Republic
dedek@ksi.mff.cuni.cz

Peter Vojtáš
Department of Software Engineering
MFF, Charles University
Prague, Czech Republic
vojtas@ksi.mff.cuni.cz

***Abstract*—Information extraction (IE) and automated semantic annotation of text are usually done by complex tools. These tools use some kind of a model that represents the actual task and its solution. The model is usually represented as a set of extraction rules (e.g., regular expressions), gazetteer lists, or it is based on some statistical measurements and probability assertions. In the environment of the Semantic Web it is essential that information is shareable and some ontology based IE tools keep the model in so called extraction ontologies. In practice, the extraction ontologies are usually strongly dependent on a particular extraction/annotation tool and cannot be used separately. In this paper, we present an extension of the idea of extraction ontologies. According to the presented concept the extraction ontologies should not be dependent on the particular extraction/annotation tool. In our solution the extraction/annotation process can be done separately by an ordinary reasoner. We also present a proof of concept for the idea: a case study with a linguistically based IE engine that exports its extraction rules to an extraction ontology and we demonstrate how this extraction ontology can be applied to a document by a reasoner. The paper also contains an evaluation experiment with several OWL reasoners.**

Keywords-Extraction Ontology; Reasoning; Information Extraction; Semantic Annotation;

I. Introduction

Information extraction (IE) and automated semantic annotation of text are usually done by complex tools and all these tools use some kind of model that represents the actual task and its solution. The model is usually represented as a set of some kind of extraction rules (e.g., regular expressions), gazetteer lists or it is based on some statistical measurements and probability assertions (classification algorithms like Support Vector Machines (SVM), Maximum Entropy Models, Decision Trees, Hidden Markov Models (HMM), Conditional Random Fields (CRF), etc.)

In the beginning, a model is either created by a human user or it is learned from a training dataset. Then, in the actual extraction/annotation process, the model is used as a configuration or as a parameter of the particular extraction/annotation tool. These models are usually stored in proprietary formats and they are accessible only by the corresponding tool.

In the environment of the Semantic Web it is essential that information is shareable and some ontology based IE tools keep the model in so called extraction ontologies [1]. Extraction ontologies should serve as a wrapper for documents of a narrow domain of interest. When we apply an extraction ontology to a document, the ontology identifies objects and relationships present in the document and it associates them with the corresponding ontology terms and thus wraps the document so that it is understandable in terms of the ontology [1].

In practice the extraction ontologies are usually strongly dependent on a particular extraction/annotation tool and cannot be used separately. The strong dependency of an extraction ontology on the corresponding tool makes it very difficult to share. When an extraction ontology cannot be used outside the tool there is also no need to keep the ontology in a standard ontology format (RDF or OWL).

The only way how to use such extraction ontology is within the corresponding extraction tool. It is not necessary to have the ontology in a "owl or rdf file". In a sense such extraction ontology is just a configuration file. For example in [2] (and also in [1]) the so called extraction ontologies are kept in XML files with a proprietary structure and it is absolutely sufficient, there is no need to treat them differently.

A. Shareable Extraction Ontologies

In this paper, we present an extension of the idea of extraction ontologies. We adopt the point that extraction models are kept in extraction ontologies and we add that the extraction ontologies should not be dependent on the particular extraction/annotation tool. In such case the extraction/annotation process can be done separately by an ordinary reasoner.

In this paper, we present a proof of concept for the idea: a case study with our linguistically based IE engine and an experiment with several OWL reasoners. In the case study (see Section IV) the IE engine exports its extraction rules to the form of an extraction ontology. Third party linguistic tool linguistically annotates an input document and the linguistic

 ISBN: 978-1-61208-175-5

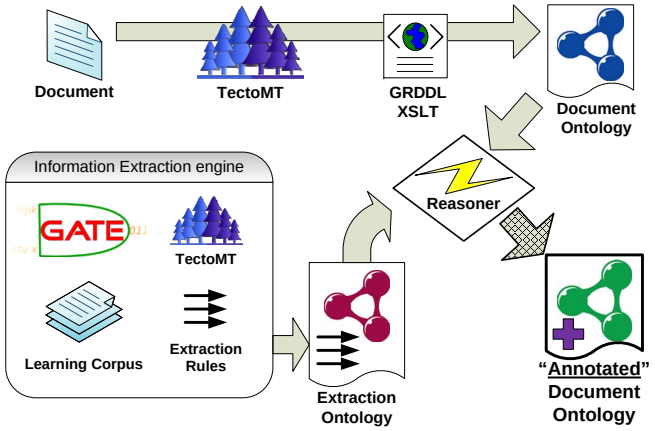


Figure 1. Semantic annotation driven by an extraction ontology and a reasoner – schema of the process.

annotations are translated to so-called document ontology. After that an ordinary OWL reasoner is used to apply the extraction ontology on the document ontology, which has the same effect as a direct application of the extraction rules on the document. The process is depicted in Fig 1 and it will be described in detail in Section IV-B.

Section II presents several closely related works. The main idea of the paper will be described in Section III, its implementation in Section IV and in Section V an experiment with several OWL reasoners and IE datasets will be presented. In Section VI related issues are discussed and Section VII concludes the paper.

II. Related Work

Ontology-based Information Extraction (OBIE) [3] or Ontology-driven Information Extraction [4] has recently emerged as a subfield of information extraction. Furthermore, Web Information Extraction [5] is a closely related discipline. Many extraction and annotation tools can be found in the above mentioned surveys ([3], [5]), many of the tools also use an ontology as the output format, but almost all of them store their extraction models in proprietary formats and the models are accessible only by the corresponding tool.

In the literature we have found only two approaches that use extraction ontologies. The former one was published by D. Embley [1], [6] and the later one – IE system Ex was developped by M. Labský [2]. But in both cases the extraction ontologies are dependent on the particular tool and they are kept in XML files with a proprietary structure.

By contrast authors of [3] (a recent survey of OBIE systems) do not agree with allowing for extraction rules to be a part of an ontology. They use two arguments against that:

1) Extraction rules are known to contain errors (because they are never 100% accurate), and objections can be raised on their inclusion in ontologies in terms of formality and accuracy.
2) It is hard to argue that linguistic extraction rules should be considered a part of an ontology while information extractors based on other IE techniques (such as SVM, HMM, CRF, etc. classifiers used to identify instances of a class when classification is used as the IE technique) should be kept out of it: all IE techniques perform the same task with comparable effectiveness (generally successful but not 100% accurate). But the techniques advocated for the inclusion of linguistic rules in ontologies cannot accommodate such IE techniques.

   The authors then conclude that either all information extractors (that use different IE techniques) should be included in the ontologies or none should be included.

Concerning the first argument, we have to take into account that extraction ontologies are not ordinary ontologies, it should be agreed that they do not contain 100% accurate knowledge. Also the estimated accuracy of the extraction rules can be saved in the extraction ontology and it can then help potential users to decide how much they will trust the extraction ontology.

Concerning the second argument, we agree that in the case of complex classification based models (SVM, HMM, CRF, etc.) serialization of such model to RDF does not make much sense (cf. the next section). But on the other hand we think that there are cases when shareable extraction ontologies can be useful and in the context of Linked Data providing shareable descriptions of information extraction rules may be valuable. It is also possible that new standard ways how to encode such models to an ontology will appear in the future.

III. Semantic Annotation Semantically

The problem of extraction ontologies that are not shareable was pointed out in the introduction (Section I). The cause of the problem is that a particular extraction model can only be used and interpreted by the corresponding extraction tool. If an extraction ontology should be shareable, there has to be a commonly used tool that is able to interpret the extraction model encoded by the extraction ontology. In this paper we present a proof of concept that Semantic Web reasoners can play the role of commonly used tools that can interpret shareable extraction ontologies.

Although it is probably always possible to encode an extraction model using a standard ontology language, only certain way of encoding makes it possible to interpret such model by a standard reasoner in the same way as if the original extraction tool was used. The difference is in semantics. It is not sufficient to encode just the model's data, it is also necessary to encode the semantics of the model. Only then the reasoner is able to interpret the model in the same way as the original tool. And this is where the title of the paper and the present section comes from. If the process of information extraction or semantic annotation should be

 ISBN: 978-1-61208-175-5

performed by an ordinary Semantic Web reasoner then only means of semantic inference are available and the extraction process must be correspondingly semantically described.

In the presented solution the approaching support for Semantic Web Rule Language (SWRL) [7] is exploited. Although SWRL is not yet approved by W3C it is already widely supported by Semantic Web tools including many OWL reasoners. The SWRL support makes it much easier to transfer the semantics of extraction rules used by our IE tool. The case study in Section IV demonstrates the translation of the native extraction rules to SWRL rules that form the core of the extraction ontology.

IV. The Main Idea Illustrated – a Case Study

In this section, realization of the main idea of the paper will be described and illustrated on a case study.

A. Document Ontologies

The main idea of this paper assumes that extraction ontologies will be shareable and they can be applied on a document outside of the original extraction/annotation tool. We further assert that the extraction ontologies can be applied by ordinary reasoners. This assumption implies that both extraction ontologies and documents have to be in a reasoner readable format. In the case of contemporary OWL reasoners there are standard reasoner-readable languages: OWL and RDF in a rich variety of possible serializations (XML, Turtle, N-Triples, etc.) Besides that there exists standard ways like GRDDL or RDFa how to obtain a RDF document from an "ordinary document" (strictly speaking XHTML and XML documents).

We call 'document ontology' an ontology that formally captures content of a document. A document ontology can be for example obtained from the source document by a suitable GRDDL transformation (as in our experiment). A document ontology should contain all relevant data of a document and preferably the document could be reconstructed from the document ontology on demand.

When a reasoner is applying an extraction ontology to a document, it only has "to annotate" the corresponding document ontology, not the document itself. Here "to annotate" means to add new knowledge – new class membership or property assertions. In fact it means just to do the inference tasks prescribed by the extraction ontology on the document ontology.

Obviously when a document can be reconstructed from its document ontology (this is very often true, it is necessary just to save all words and formatting instructions) then also an annotated document can be reconstructed from its annotated document ontology.

B. Implementation

In this section, we will present details about the case study. We have used our IE engine [8] based on deep linguistic parsing and Inductive Logic Programming. It is a complex system implemented with a great help of the GATE system (http://gate.ac.uk/) and it also uses many other third party tools including several linguistic tools and a Prolog system. Installation and making the system operate is not simple. This case study should demonstrate that the extraction rules produced by the system are not dependent on the system in the sense described above.

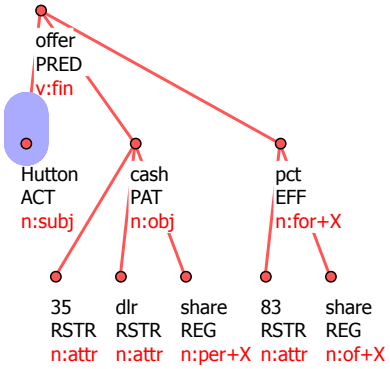


Figure 2. Tectogrammatical tree of the sentence: "Hutton is offering 35 dlrs cash per share for 83 pct of the shares." Nodes roughly correspond with words of a sentence, edges represent linguistic dependencies between nodes and some linguistic features (tectogrammatical lemma, semantic functor and semantic part of speech) are printed under each node.

1) Linguistic Analysis: Our IE engine needs a linguistic preprocessing (deep linguistic parsing) of documents on its input. Deep linguistic parsing brings a very complex structure to the text and the structure serves as a footing for construction and application of extraction rules.

We usually use TectoMT system [9] to do the linguistic preprocessing. TectoMT is a Czech project that contains many linguistic analyzers for different languages including Czech and English. We are using a majority of applicable tools from TectoMT: a tokeniser, a sentence splitter, morphological analyzers (including POS tagger), a syntactic parser and the deep syntactic (tectogrammatical) parser. All the tools are based on the dependency based linguistic theory and formalism of the Prague Dependency Treebank (PDT, http://ufal.mff.cuni.cz/pdt2.0/).

The output linguistic annotations of the TectoMT system are stored (along with the text of the source document) in XML files in so called Prague Markup Language (PML, http://ufal.mff.cuni.cz/jazz/PML/). PML is a very complex language (or XML schema) that is able to express many linguistic elements and features present in text. For the IE engine a tree dependency structure of words in sentences is the most useful one because the edges of the structure guide the extraction rules. An example of such (tectogrammatical) tree structure is in Fig. 2.

In this case study, PML files made from source documents by TectoMT are transformed to RDF document ontology by quite simple GRDDL/XSLT transformation. Such document ontology contains the whole variety of PML in RDF format.

 ISBN: 978-1-61208-175-5

```
[Rule 1] [Pos cover = 23 Neg cover = 6]
mention_root(acquired,A) :-
   'lex.rf'(B,A), t_lemma(B,'Inc'),
   tDependency(C,B), tDependency(C,D),
   formeme(D,'n:in+X'), tDependency(E,C).
[Rule 11] [Pos cover = 25 Neg cover = 6]
mention_root(acquired,A) :-
   'lex.rf'(B,A), t_lemma(B,'Inc'),
   tDependency(C,B), formeme(C,'n:obj'),
   tDependency(C,D), functor(D,'APP').
[Rule 75] [Pos cover = 14 Neg cover = 1]
mention_root(acquired,A) :-
   'lex.rf'(B,A), t_lemma(B,'Inc'),
   functor(B,'APP'), tDependency(C,B),
   number(C,pl).
```

Figure 3. Examples of extraction rules in the native Prolog format.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Ontology [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY pml "http://ufal.mff.cuni.cz/pdt/pml/" >
]>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
  ontologyIRI="http://czsem.berlios.de/onto ... rules.owl">
  <DLSafeRule>
    <Body>
      <ObjectPropertyAtom>
        <ObjectProperty IRI="&pml;lex.rf" />
        <Variable IRI="urn:swrl#b" />
        <Variable IRI="urn:swrl#a" />
      </ObjectPropertyAtom>
...
      <DataPropertyAtom>
        <DataProperty IRI="&pml;number" />
        <Variable IRI="urn:swrl#c" />
        <Literal>pl</Literal>
      </DataPropertyAtom>
    </Body>
    <Head>
      <DataPropertyAtom>
        <DataProperty IRI="&pml;mention_root" />
        <Literal>acquired</Literal>
        <Variable IRI="urn:swrl#a" />
      </DataPropertyAtom>
    </Head>
  </DLSafeRule>
</Ontology>
```

Figure 4. Rule 75 in the OWL/XML syntax for Rules in OWL 2 [10].

```
@prefix pml: <http://ufal.mff.cuni.cz/pdt/pml/>.
[rule-75:
        ( ?b pml:lex.rf ?a )
        ( ?c pml:tDependency ?b )
        ( ?b pml:functor 'APP' )
        ( ?c pml:number 'pl' )
        ( ?b pml:t_lemma 'Inc' )
     ->
        ( ?a pml:mention_root 'acquired' )
]
```

Figure 5. Rule 75 in the Jena rules syntax.

2) Rule Transformations: Extraction rules produced by the IE engine are natively kept in a Prolog format; examples can be seen in Fig. 3. The engine is capable to export them to the OWL/XML syntax for rules in OWL 2 [10] (see in Fig. 4). Such rules can be parsed by OWL API (http://owlapi.sourceforge.net/) 3.1 and exported to RDF/SWRL, which is very widely supported and hopefully becoming a W3C recommendation. The last rule example can be seen in Fig. 5, it shows a rule in the Jena rules format. Conversion to Jena rules was necessary because it is the only format that Jena can parse, see details about our use of Jena in Section V. The Jena rules were obtained using following transformation process: OWL/XML → RDF/SWRL conversion using OWL API and RDF/SWRL → Jena rules conversion using SweetRules (http://sweetrules.semwebcentral.org/).

The presented rules belong to the group of so called DL-Safe rules [11] so the decidability of OWL reasoning is kept.

3) Schema of the Case Study: A schema of the case study was presented in Fig. 1. The top row of the image illustrates how TectoMT (third party linguistic tool) linguistically annotates an input document and the linguistic annotations are translated to so-called document ontology by a GRDDL/XSLT transformation.

In the bottom of the picture our IE engine learns extraction rules and exports them to an extraction ontology. The reasoner in the middle is used to apply the extraction ontology on the document ontology and it produces the "annotated" document ontology, which was described in Section IV-A.

V. Experiment

In this section, we present an experiment that should serve as a proof of a concept that the proposed idea of independent extraction ontologies is realizable. We have selected several reasoners (namely Jena, HermiT, Pellet and FaCT++) and tested them on two slightly different datasets from two different domains and languages (see Table I). This should at least partially demonstrate the universality of the proposed approach.

In both cases the task is to find all instances (corresponding to words in a document) that should be uncovered by the extraction rules. The extraction rules are saved in single extraction ontology for each dataset. The datasets are divided into individual document ontologies (owl files) corresponding to the individual documents. During the experiment the individual document ontologies are processed separately (one ontology in a step) by a selected reasoner. The total time taken to process all document ontologies of a dataset is the measured result of the reasoner for the dataset.

The actual reasoning tasks are more difficult than a simple retrieval of all facts entailed by the extraction rules. Such simple retrieval task took only a few seconds for the Acquisitions v1.1 dataset (including parsing) in the native Prolog environment that the IE engine uses. There were several more inferences needed in the reasoning tasks because the schema of the input files was a little bit different from the schema used in rules. The mapping of the schemas was captured in another "mapping" ontology that was included in the reasoning. The mapping ontology is a part of the publically available project ontologies.

A. How to Download

All the resources (including source codes of the case study and the experiment, datasets and ontologies) mentioned in this paper are publically available on the project's web site (http://czsem.berlios.de/ (before 2012) or http://czsem.

 ISBN: 978-1-61208-175-5

Table I
DESCRIPTION OF DATASETS THAT WERE USED.

dataset	domain	language	number of files	dataset size (MB)	number of rules
czech_fireman	accidents	Czech	50	16	2
acquisitions	finance	English	600	126	113

sourceforge.net/) and detailed information can be found there.

B. Datasets

In the experiment we used two slightly different datasets from two different domains and languages. Table I summarizes some basic information about them.

1) Czech Fireman: The fist dataset is called 'czech_fireman'. This dataset was created by ourselves during the development of our IE engine. It is a collection of 50 Czech texts that are reporting on some accidents (car accidents and other actions of fire rescue services). These reports come from the web of Fire rescue service of Czech Republic. The corpus is structured such that each document represents one event (accident) and several attributes of the accident are marked in text. For the experiment we selected the 'damage' task – to find an amount (in CZK - Czech Crowns) of summarized damage arisen during a reported accident.

2) Acquisitions v1.1: The second dataset is called "Corporate Acquisition Events". More precisely we used the *Acquisitions v1.1* version[1] of the corpus. This is a collection of 600 news articles describing acquisition events taken from the Reuters dataset. News articles are tagged to identify fields related to acquisition events. These fields include 'purchaser' , 'acquired', and 'seller' companies along with their abbreviated names ('purchabr', 'acqabr' and 'sellerabr'). Some news articles also mention the field 'deal amount'. For the experiment we selected only the 'acquired' task.

C. Reasoners

In the experiment we used four OWL reasoners:

Jena (http://jena.sourceforge.net),
HermiT (http://hermit-reasoner.com),
Pellet (http://clarkparsia.com/pellet),
FaCT++ (http://code.google.com/p/factplusplus) .

We measured the time they spent on processing a particular dataset. The time also includes time spent on parsing the input. HermiT, Pellet and FaCT++ were called through OWL API-3.1, so the same parser was used for them. Jena reasoner was used in its native environment with the Jena parser.

In the early beginning of the experiment we had to exclude the FaCT++ reasoner from both tests. It turned out that FaCT++ does not work with rules [12] and it did not return any result instances. All the remaining reasoners strictly agreed on the results and returned the same sets of instances.

[1]This version of the corpus comes from the Dot.kom project's resources (http://nlp.shef.ac.uk/dot.kom/resources.html 2011-08-09 page, 2006-12-31 dataset).

Table II
TIME PERFORMANCE OF TESTED REASONERS ON BOTH DATASETS.

reasoner	**czech_fireman**	stdev	**acquisitions-v1.1**	stdev
Jena	161 s	0.226	1259 s	3.579
HermiT	219 s	1.636	$\gg$ 13 hours	
Pellet	11 s	0.062	503 s	4.145
FaCT++	Does not support rules.			

Time is measured in seconds. Average values from 6 measurements. Experiment environment: Intel Core I7-920 CPU 2.67GHz, 3GB of RAM, Java SE 1.6.0_03, Windows XP.

Also HermiT was not fully evaluated on the Acquisitions v1.1 dataset because it was too slow. The reasoner spent 13 hours of running to process only 30 of 600 files of the dataset. And it did not seem useful to let it continue.

D. Evaluation Results of the Experiment

Table II summarizes results of the experiment. The standard deviations are relatively small when compared to the differences between the average times. So there is no doubt about the order of the tested reasoners. Pellet performed the best and HermiT was the slowest amongst the tested and usable reasoners in this experiment.

From the results we can conclude that similar tasks can be satisfactorily solved by contemporary OWL reasoners because three of four tested reasoners were working correctly and two reasoners finished in bearable time.

On the other hand even the fastest system took 8.5 minutes to process 113 rules over 126MB of data. This is clearly significantly longer than a bespoke system would require. Contemporary Semantic Web reasoners are known still to be often quite inefficient and the experiment showed that using them today to do information extraction will result in quite poor performance. However, efficiency problems can be solved and in the context of Linked Data providing shareable descriptions of information extraction rules may be valuable.

VI. DISCUSSION

In this paper (Section IV-A), we have described a method how to apply an extraction ontology to a document ontology and obtain so called "annotated" document ontology. To have an "annotated" document ontology is almost the same as to have an annotated document. An annotated document is useful (easier navigation, faster reading and lookup of information, possibility of structured queries on collections of such documents, etc.) but if we are interested in the actual information present in the document, if we want to know the facts that are in a document asserted about the real word things then an annotated document is not sufficient. But the conversion of an annotated document to the real world facts is not simple. There are obvious issues concerning data integration and duplicity of information. For example when in a document two mentions of people are annotated as 'injured', what is then the number of injured people in the corresponding accident? Are the two annotations in fact linked to the same person or not?

 ISBN: 978-1-61208-175-5

In the beginning of our work on the idea of shareable extraction ontologies we planned to develop it further, we wanted to cover also the step from annotated document ontologies to the real world facts. The extraction process would then end up with so called "fact ontologies". But two main obstacles prevent us to do that.

1) Our IE engine is not yet capable to solve these data integration and duplicity of information issues and the real world facts would be quite imprecise then.
2) There are also technology problems of creating new facts (individuals) during reasoning.

A. SPARQL Queries – Increasing Performance?

There is also a possibility to transform the extraction rules to SPARQL construct queries. This would probably rapidly increase the time performance. However a document ontology would then have to exactly fit with the schema of the extraction rules. This would be a minor problem.

The reason why we did not study this approach from the beginning is that we were interested in extraction *ontologies* and SPARQL queries are not currently regarded as a part of an ontology and nothing is suggesting it to be other way round. Anyway the performance comparison remains a valuable task for the future work.

B. Contributions for Information Extraction

The paper combines the field of ontology-based information extraction and rule-based reasoning. The aim is to show a new possibility in usage of IE tools and reasoners. In this paper, we do not present a solution that would improve the performance of IE tools.

We also do not provide a proposal of a universal extraction format (although a specific form for the rule based extraction on dependency parsed text could be inferred). This task is left for the future if a need for such activity emerges.

VII. Conclusion

In the beginning of the paper we pointed out the draw back of so called extraction ontologies – in most cases they are dependent on a particular extraction/annotation tool and they cannot be used separately.

We extended the concept of extraction ontologies by adding the shareable aspect and we introduced a new principle of making extraction ontologies independent of the original tool: the possibility of application of an extraction ontology to a document by an ordinary reasoner.

In Section IV we presented a case study that shows that the idea of shareable extraction ontologies is realizable. We presented implementation of an IE tool that exports its extraction rules to an extraction ontology and we demonstrated how this extraction ontology can be applied to a document by a reasoner. Moreover, in Section V, an experiment with several OWL reasoners was presented. The experiment evaluated the performance of contemporary OWL reasoners on IE tasks (application of extraction ontologies).

A new publically available benchmark for OWL reasoning was created together with the experiment. Other reasoners can be tested this way.

Acknowledgments: This work was partially supported by Czech projects: GACR P202/10/0761, GACR-201/09/H057, GAUK 31009 and MSM-0021620838.

References

[1] D. W. Embley, C. Tao, and S. W. Liddle, "Automatically extracting ontologically specified data from html tables of unknown structure," in *ER*, ser. LNCS, vol. 2503. Springer, 2002, pp. 322–337.

[2] M. Labský et al., "The Ex Project: Web Information Extraction Using Extraction Ontologies," in *Knowledge Discovery Enhanced with Semantic and Social Information*, ser. Studies in Comput. Intellig. Springer, 2009, vol. 220, pp. 71–88.

[3] D. C. Wimalasuriya and D. Dou, "Ontology-based information extraction: An introduction and a survey of current approaches," *Journal of Information Science*, vol. 36, no. 3, pp. 306–323, June 2010.

[4] B. Yildiz and S. Miksch, "ontoX - a method for ontology-driven information extraction," in *ICCSA'07 - Part III*. Springer, 2007, pp. 660–673.

[5] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan, "A survey of web information extraction systems," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1411–1428, 2006.

[6] D. W. Embley, "Toward semantic understanding: an approach based on information extraction ontologies," in *ADC '04*. Darlinghurst: ACS, 2004, pp. 3–12.

[7] B. Parsia, E. Sirin, B. C. Grau, E. Ruckhaus, and D. Hewlett. (2005) Cautiously approaching SWRL. Preprint submitted to Elsevier Science. 2011-08-09. [Online]. Available: http://www.mindswap.org/papers/CautiousSWRL.pdf

[8] J. Dědek, "Towards semantic annotation supported by dependency linguistics and ILP," in *ISWC2010*, ser. LNCS, vol. 6497. Springer, 2010, pp. 297–304.

[9] Z. Žabokrtský, J. Ptáček, and P. Pajas, "TectoMT: Highly modular MT system with tectogrammatics used as transfer layer," in *Proceedings of the 3rd Workshop on Statistical Machine Translation*. Columbus, OH, USA: ACL, 2008, pp. 167–170.

[10] B. Glimm, M. Horridge, B. Parsia, and P. F. Patel-Schneider, "A Syntax for Rules in OWL 2," in *OWLED 2009*, vol. 529. CEUR, 2009.

[11] B. Motik, U. Sattler, and R. Studer, "Query answering for owl-dl with rules," *Web Semantics*, vol. 3, pp. 41–60, July 2005.

[12] Wikipedia article: Semantic reasoner - reasoner comparison. 2011-08-09. [Online]. Available: http://en.wikipedia.org/wiki/Semantic_reasoner#Reasoner_comparison

 ISBN: 978-1-61208-175-5

Toward the Automatic Generation of a Semantic VRML Model from Unorganized 3D Point Clouds

Helmi Ben Hmida, Frank Boochs
Institut i3mainz, am Fachbereich Geoinformatik und Vermessung
Fachhochschule Mainz, Lucy-Hillebrand-Str. 255128 Mainz, Germany
e-mail: {helmi.benhmida, boochs}@geoinform.fh-mainz.de

Christophe Cruz, Christophe Nicolle
Laboratoire Le2i, UFR Sciences et Techniques
Université de Bourgogne
B.P. 47870, 21078 Dijon Cedex, France
e-mail: {christophe.cruz, cnicolle}@u-bourgogne.fr

Abstract—This paper presents our experience regarding the creation of 3D semantic facility model out of unorganized 3D point clouds. Thus, a knowledge-based detection approach of objects using the OWL ontology language is presented. This knowledge is used to define SWRL detection rules. In addition, the combination of 3D processing built-ins and topological Built-Ins in SWRL rules aims at combining geometrical analysis of 3D point clouds and specialist's knowledge. This combination allows more flexible and intelligent detection and the annotation of objects contained in 3D point clouds. The created WiDOP prototype takes a set of 3D point clouds as input, and produces an indexed scene of colored objects visualized within VRML language as output. The context of the study is the detection of railway objects materialized within the Deutsche Bahn scene such as signals, technical cupboards, electric poles, etc. Therefore, the resulting enriched and populated domain ontology, that contains the annotations of objects in the point clouds, is used to feed a GIS system.

Keywords-Semantic facility information model; Semantic VRML model; Geometric analysis; Topological analysis; 3D processing algorithm; Semantic web; knowledge modeling; ontology; 3D scene reconstruction; object identification.

I. INTRODUCTION

The technical survey of facility aims to build a digital model based on geometric analysis. Such a process becomes more and more tedious, especially with the generation of the new terrestrial laser scanners, faster, accurate, where huge amount of 3D point clouds is generated. Within such new technologies, new challenges have seen the light where the basic one is to make the reconstruction process automatic and more accurate. Thus, early works on 3D point clouds have investigated the reconstruction and the recognition of geometrical shapes [1], [2]. This problematic was investigated as a topic of the computer graphic and the signal processing research where most works focused on segmentation or visualization aspects. As most recent works, new tendency related to the use of semantic has been explored [3]. In fact, we agree with the assumption that it helps the improvement of the automation, the accuracy and the result quality, but we think that it has to be well studied and proved. Otherwise, how the detection process can get support within different knowledge about the scene objects and what´s its impact compared to classic approach. In such scenario, knowledge about such objects has to include detailed information about the objects geometry, structure, 3D algorithms, etc.

By this contribution, we suggest a solution to the problematic of facility survey model creation from 3D point clouds with knowledge support. The suggested system is materialized via WiDOP project [4]. Furthermore, the created WiDOP platform is able to generate an indexed scene from unorganized 3D point clouds visualized within virtual reality modeling language [5].

The reminder of this paper is organized as follows: The next section describes briefly the most recent related works, followed by the prototype definition in section three. In section four, more focus on the domain ontology structure presenting the core behind WiDOP prototype will take place where we highlight the ontology structure and the created extension with the SWRL language to satisfy the target purpose. Section five presents a use cased materialized by the scene of the German rail way. Finally, we conclude and give insight on our future work in section six.

II. BACKGROUND CONCEPT AND METHODOLOGY

The technical survey of facilities, as a long and costly process, aims at building a digital model based on geometric analysis since the modeling of a facility as a set of vectors is not sufficient in most cases. To resolve this problem a new standard was developed over ten years by the International Alliance for Interoperability (IAI). This standard, called IFC [6], considers the facility elements as objects that are defined by a 3D geometry and normalized semantic [14]. The problematic of 3D object detection and scene reconstruction including semantic knowledge was recently treated within different domain, basically the photogrammetry one [7], the construction one, the robotics [8] and recently the knowledge engineering one [4]. Modeling a survey, in which low-level point cloud or surface representation is transformed into a semantically rich model is done in three tasks where the first is the data collection, in which dense point measurements of the facility are collected using laser scans taken from key locations throughout the facility; Then data processing, in which the sets of point clouds from the collected scanners are processed. Finally, modeling the survey in which the low-level point cloud is transformed into a semantically rich model. This is done via modeling geometric knowledge, qualifying topological relations and finally assigning an object category to each geometry [9].

 ISBN: 978-1-61208-175-5

Concerning the geometry modeling, we remind here that the goal is to create simplified representations of facility components by fitting geometric primitives to the point cloud data [17]. The modeled components are labeled with an object category. Establishing relationships between components is important in a facility model and must also be established. In fact, relationships between objects in a facility model are useful in many scenarios. In addition, spatial relationships between objects provide contextual information to assist in object recognition [10]. Within the literature, three main strategies are described to rich such a model where the first one is based on human interaction with provided software's for point clouds classifications and annotations [11]. While the second strategy relies more on the automatic data processing without any human interaction by using different segmentation techniques for feature extraction [8]. Finally, new techniques presenting an improvement compared with the cited ones by integrating semantic networks to guide the reconstruction process [12].

A. Manual survey model creation

In current practice, the creation of facility model is largely a manual process performed by service providers who are contracted to scan and model a facility. In reality, a project may require several months to be achieved, depending on the complexity of the facility and the modeling requirements. Reverse engineering tools excel at geometric modeling of surfaces, but with lack of volumetric representations, while such design systems cannot handle the massive data sets from laser scanners. As a result, modelers often shuttle intermediate results back and forth between different software packages during the modeling process, giving rise to the possibility of information loss due to limitations of data exchange standards or errors in the implementation of the standards within the software tools [13]. Prior knowledge about component geometry, such as the diameter of a column, can be used to constrain the modeling process, or the characteristics of known components may be kept in a standard component library. Finally, the class of the detected geometry is determined by the modeler once the object created. In some cases, relationships between components are established either manually or in a semi-automated manner.

B. Semi-Automatic and Automatic methods

The manual process for constructing a survey model is time consuming, labour-intensive, tedious, subjective, and requires skilled workers. Even if modeling of individual geometric primitives can be fairly quick, modeling a facility may require thousands of primitives. The combined modeling time can be several months for an average sized facility. Since the same types of primitives must be modeled throughout a facility, the steps are highly repetitive and tedious [12]. The above mentioned observations and others illustrate the need for semi-automated and automated techniques for facility model creation. Ideally, a system could be developed that would take a point cloud of a facility as input and produce a fully annotated as-built model of the facility as output. The first step within the automatic process is the geometric modeling. It presents the process of constructing simplified representations of the 3D shape of survey components from point cloud data. In general, the shape representation is supported by CSG [15] or B-Rep [16] representation. The representation of geometric shapes has been studied extensively [15]. Once geometric elements are detected and stored via a specific presentation, the final task within a facility modeling task is the object recognition. It presents the process of labeling a set of data points or geometric primitives extracted from the data with a named object or object class. Whereas the modeling task would find a set of points to be a vertical plane, the recognition task would label that plane as being a wall, for instance. Often, the knowledge describing the shapes to be recognized is encoded in a set of descriptors that implicitly capture object shape. Research on recognition of facilities specific components related to a facility is still in its early stages. Methods in this category typically perform an initial shape-based segmentation of the scene, into planar regions, for example, and then use features derived from the segments to recognize objects. This approach is exemplified by Rusu et al. who use heuristics to detect walls, floors, ceilings, and cabinets in a kitchen environment [8]. A similar approach was proposed by Pu and Vosselman to model facility façades [18].

To reduce the search space of object recognition algorithms, the use of knowledge related to a specific facility can be a fundamental solution. For instance, Yue et al. overlay a design model of a facility with the as-built point cloud to guide the process of identifying which data points belong to specific objects and to detect differences between the as-built and as-designed conditions [19]. In such cases, object recognition problem is simplified to be a matching problem between the scene model entities and the data points. Another similar approach is presented in [20]. Other promising approaches have only been tested on limited and very simple examples, and it is equally difficult to predict how they would fare when faced with more complex and realistic data sets. For example, the semantic network methods for recognizing components using context work well for simple examples of hallways and barren, rectangular rooms [10], but how would they handle spaces with complex geometries and clutter.

C. Discussion:

The presented methods for survey modeling and object recognition rely on hand-coded knowledge about the domain. Concepts like "Signals are vertical" and "Signals intersect with the ground" are encoded within the algorithms either explicitly, through sets of rules, or implicitly, through the design of the algorithm. Such hard-coded, rule based approaches tend to be brittle and break down when tested in new and slightly different environments. Furthermore, it can

 ISBN: 978-1-61208-175-5

be difficult to extend an algorithm with new rule or to modify the rules to work in new environments. Based on these observations, we predict that more standard and flexible representations of facility objects and more sophisticated guidance based algorithms for object detection instead of a standard one will open the way to significant improvement in facility modeling capability and generality.

III. WiDOP PROTOTYPE

WiDOP platform is a Java platform presenting a knowledge based detection of objects in point clouds based on OWL ontology language, Semantic Web Rule Language, and 3D processing algorithms. It aims at combining geometrical analysis of 3D point clouds and specialist's knowledge to get a more reliable facility model. In fact, this combination allows the detection and the annotation of objects contained in point clouds. WiDOP prototype takes in consideration the adjustment of the old methods and, in the meantime, profit from the advantages of the emerging cutting edge technology. From the principal point of view, our system still retains the storing mechanism within the existent 3D processing algorithms; in addition, suggest a new field of detection and annotation, where we are getting a real-time support from the target scene knowledge. Add to that, we suggest a collaborative Java Platform based on semantic web technology (OWL, RDF, and SWRL) and knowledge engineering in order to handle the information provided from the knowledge base and the 3D packages results.

The field of the Deutsch Bahn railway scene is treated for object detection. The objective of the system consists in creating, from a set of point cloud files, from an ontology that contains knowledge about the DB railway objects and 3D processing algorithms, an automatic process that produces as output a set of tagged elements contained in the point clouds.

The process enriches and populates the ontology with new individuals and relationships between them. In order to graphically represent these objects within the scene point clouds, a VRML model file [5] is generated and visualized within the prototype where the color of objects in the VRML file represents its semantic definition. The resulting ontology contains enough knowledge to feed a GIS system, and to generate IFC file [6] for CAD software. As seen in Figure 1, the created system is composed of three parts.

- Generation of a set of geometries from a point could file based on the target object characteristics
- Computation of business rules with geometry, semantic and topological constrains in order to annotate the different detected geometries.
- Generation of a VRML model related to the scene within the detected and annotated elements

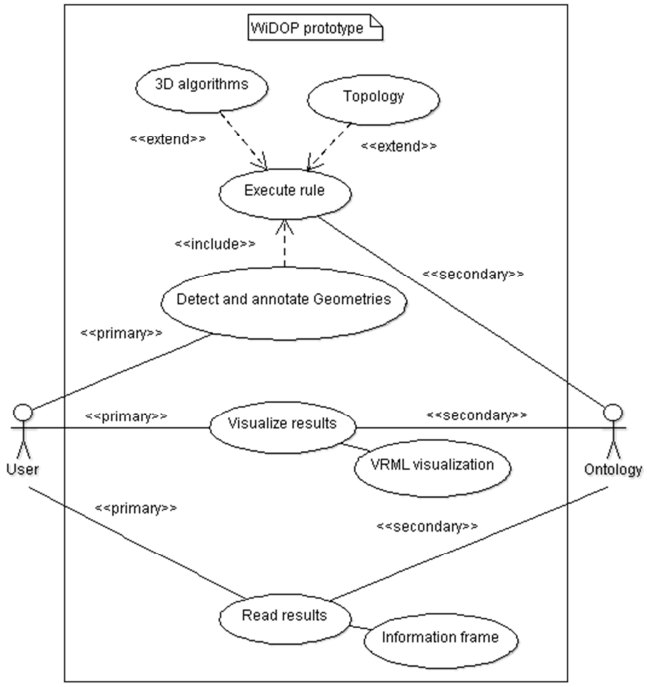


Figure 1. the WiDOP use case diagram

To rich such a target, three main steps aim at detecting and identifying objects are established:

- From 3D point clouds to geometric elements.
- From geometry to topological relations.
- From geometric and/or topological relations to semantic elements annotation.

As a first impression, the system responds to the target requirement since it would take a point cloud of a facility as input and produce a fully annotated as-built model of the facility as output. In the next, we focus on the core of the WiDOP prototype which is materialized via an ontology base structure to guide the 3D scene reconstruction process.

IV. ONTOLOGY BASED PROTOTYPE

In recent years, formal ontology has been suggested as a solution to the problem of 3D objects reconstruction from 3D point clouds [21]. In this area, ontology structure was defined as a formal representation of knowledge by a set of concepts within a domain, and the relationships between those concepts. It is used to reason about the entities within that domain, and may be used to describe the domain. Conventionally, ontology presents a "formal, explicit specification of a shared conceptualization" [22].
Well-made ontology owns a number of positive aspects like the ability to define a precise vocabulary of terms, the ability to inherit and extends exiting ones, the ability to declare relationships between defined concepts and finally the ability to infer new relationship by reasoning on existing ones. Through the scientific community, the basic strength

 ISBN: 978-1-61208-175-5

of formal ontology is their ability to reason in a logical way based on Description Logics DL. The last one presents a form of logic to reason on objects. In fact, despite the richness of OWL's set of relational properties, the axiom does not cover the full range of expressive possibilities for object relationships that we might find. For that, it is useful to declare a relationship in term of conditions or even rules. Some of the evolved languages are related to the semantic web rule language (SWRL) and advanced Jena rules [23]. SWRL is a proposal as a Semantic Web rules language, combining sublanguages of the OWL Web Ontology Language with the Rule Markup Language [24].

A. Ontology schema

This section discusses the different aspects related to the Deutsche Bahn scene ontology structure installed behind the WiDOP Deutsche Bahn prototype [4]. The domain ontology presents the core of WiDOP project and provides a knowledge base to the created application. The global schema of the modeled ontology structure offers a suitable framework to characterize the different Deutsche Bahn elements from the 3D processing point of view. The created ontology is used basically for two purposes:

- To guide the processing algorithm sequence creation based on the target object characteristics.
- To facilitate the semantic annotation of the different detected objects inside the target scene.

The created knowledge base related to the Deutsche Bahn scene has been inspired next to our discussion with the domain expert and next to our study based on the official Web site for the German rail way specification [25]. The current ontology is divided onto three main parts: the Deutsche Bahn concepts, the algorithm concepts and the geometry concepts. However, they will be used with others to facilitate the object detection based on SWRL and the automatic annotation of Bounding Box geometry based on inference engine tools. At this level, no real interaction between human and the knowledge base is taken in consideration, since the 3D detection process algorithm and parameters are alimented directly from the knowledge base and then interpreted by the SWRL rules and Description Logics tools. The ontology is managed through different components of Description Logics. There are five main classes within other data and objects properties able to characterize the scene in question.

- Algorithm
- Geometry
- DomainConcept
- Characteristics
- Scene

The class DomainConcept can be considered the main class in this ontology as it is the class where the different elements within a 3D scene are defined. It was designed after the DB scene observation. It contains all kinds of elements, which have to be detected and is divided in two general classes, one for the Furniture and one for the Facility Element. However, the importance of other classes cannot be ignored. They are used to either describe the domain concept geometry and characteristics or to define the 3D processing algorithms within the target geometry. The subclasses of the Algorithm class represent the different developed algorithms. They are related to several properties which are able to detect. These properties (Geometric and semantic) are shared with the DomainConcept and the Geometry classes. By this way, a created sequence of algorithms can detect all the characteristics of an element while the Geometry class represents every kind of geometry, which can be detected in the point cloud scene.

The connection between the basic mentioned classes is carried out through object and data properties. There exist object properties for each mentioned activities. Besides, the object properties are also used to relate an object to other objects via topological relations. In general, there are five general object properties in the ontology which have their specialized properties for the specialized activities. They are

- hasTopologicRelation
- IsDeseignedFor
- hasGeometry
- hasCharacteristics,

Figure **2** demonstrates the general layout schema of the application.

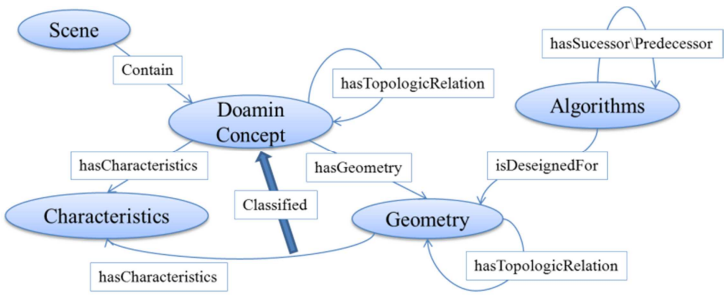


Figure 2. Ontology general schema overview

B. Enrichment of the ontology within processing and topologic operations

To support the defined use cases, two basic further layers to the semantic one are added to ontology in order to ensure the geometry detection and annotation process tasks. These operations are the 3D processing and topological relations qualification respectively.

1) 3D processing operations

The 3D processing layer contains all relevant aspects related to the 3D processing algorithms. Its integration into the WiDOP semantic framework is done by special Built-Ins. They manage the interaction between the above mentioned

layers and the semantic one. In addition, it contains the different algorithm definitions, properties, and the related geometries to the each defined algorithms. An important achievement is the detection and the identification of objects with specific characteristics such as a signal, indicator columns, and electric pole, etc. through utilizing their geometric properties. Since the information in point cloud data sometimes is unclear and insufficient.

2) Topological operations

The layer of the topological knowledge represents topological relationships between scene elements since the object properties are also used to link an object to others by a topological relation. For instance, a topological relation between a distant signal and a main one can be defined, as both have to be distant from one kilometer. The qualification of topological relations into the semantic framework is done by topological Built-Ins.

C. Extension of SWRL with 3D processing and topological operations

This section resumes the adopted approach to integrate the mentioned processing and topologic operation with help of the swrl language (Horn clauses) in order to define new knowledge (Classes and properties) related to the as built facility modeling. We recall that SWRL Built-ins allow further extensions within a defined taxonomy. In fact, it helps in the interoperation of SWRL with other formalisms by providing an extensible, modular built-ins infrastructure for Semantic Web Languages and knowledge based applications. For such a reason, we opt to be based on such a technology to extend the actual knowledge base within two basic Built-Ins: Topologic Built-Ins and Processing Built-Ins.

1) Extension of standard SWRL with processing operations

The first step aims at the geometric elements' detection. Thus, Semantic Web Rule Language within extended built-ins is used to execute a real 3D processing algorithm first, and to populate the provided knowledge within the ontology (e.g., Table 1). The "3D_swrlb_Processing: VerticalElementDetection" built-ins for example, aims at the detection of geometry with vertical orientation. The prototype of the designed Built-in is:

```
3D_swrlb_Processing:VerticalElementDetection
(?Vert, ?Dir)
```

where the first parameter presents the target object class, and the last one presents the point clouds' directory defined within the created scene in the ontology structure. At this point, the detection process will result bounding boxes, representing a rough position and orientation of the detected object. Table 1 show the mapping between the 3D processing built-ins, which is computer and translated to predicate, and the corresponding class.

TABLE 1. 3D PROCESSING BUILT-INS MAPPING PROCESS

3D Processing Built-Ins	Correspondent Simple class
3D_swrlb_Processing: VerticalElementDetection (?Vert,?Dir)	Vertical_BoundingBox(?x)
3D_swrlb_Processing: HorizentalElementDetection (?Vert,?Dir)	Horizental_BoundingBox(?y)

2) Extension of standard SWRL with topologic operations

Once geometries are detected, the second step, aims at verifying certain topology properties between detected geometries. Thus, 3D_Topologic built-ins have been added in order to extend the SWRL language. Topological rules are used to define constrains between different elements. After parsing the topological built-ins and its execution, the result is used to enrich the ontology with relationships between individuals that verify the rules. Similarly, to the 3D processing built-ins, our engine translates the rules with topological built-ins to standard rules, Table 2.

TABLE 2. EXAMPLE OF TOPOLOGICAL BUILT-INS

Processing Built-Ins	Correspondent object property
3D_swrlb_Topology:Upper(?x, ?y)	Upper(?x,?y)
3D_swrlb_Topology:Intersect(?x, ?y)	Intersect (?x,?y)

V. CASE STUDY

For the demonstration of our prototype, 500 m from the scanned point clouds related to Deutsch Bahn scene in the city of Nürnberg was extracted. It contains a variety of the target objects. The whole scene has been scanned using a terrestrial laser scanner fixed within a train, resulting in a large point cloud representing the surfaces of the scene objects. Within the created prototype, different rules are processed, (see Figure 3). First, geometrical elements will be searched in the area of interest based on dynamic 3D processing algorithm sequence created based on semantic object properties, and then topological relations between detected geometries are qualified. Subsequently, further annotation may be relayed on aspects expressing facts to orientation or size of elements, which may be sufficient to finalize a decision upon the semantic of an object or on a fact expressing topological relationship or both of them.

This second step within our approach aims to identify existing topologies between the detected geometries. To do, useful topologies for geometry annotation are tested. Topological Built-Ins like `isConnected`, `touch`, `Perpendicular`, `isDistantfrom` are created. As a result, relations found between geometric elements are

 ISBN: 978-1-61208-175-5

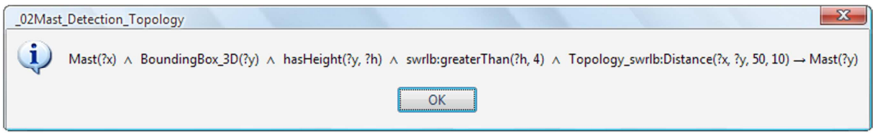


Figure 3. WiDOP prototype and example of used swrl rules within Built-Ins extention

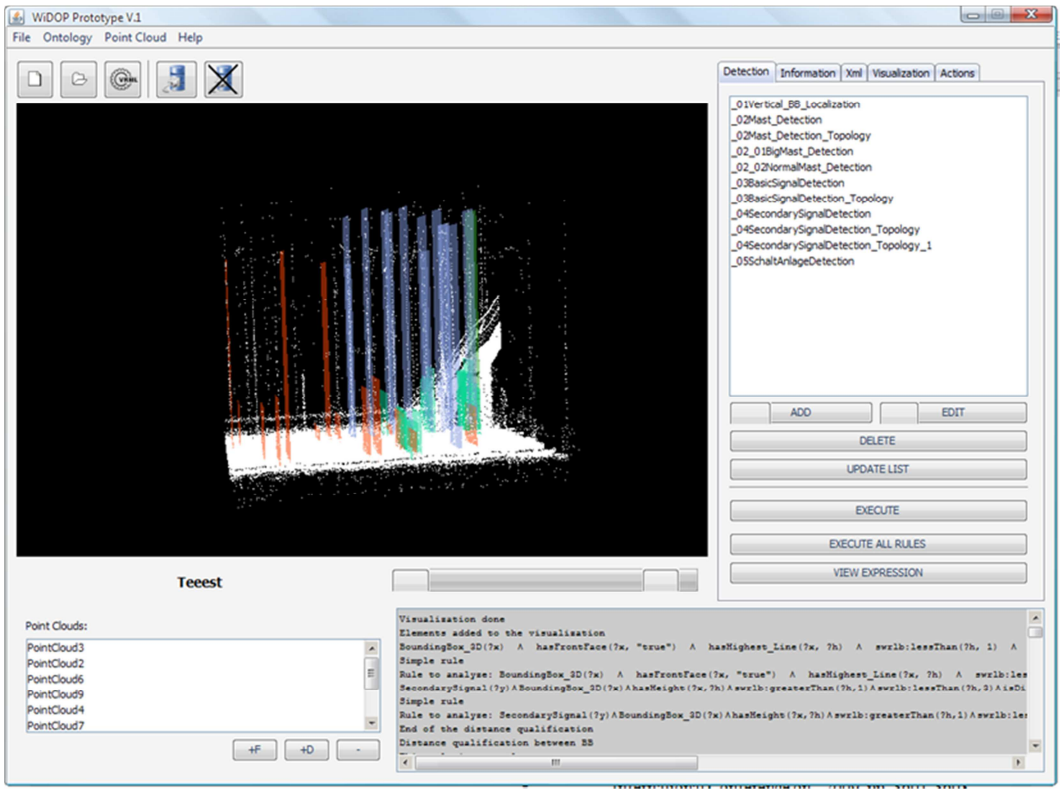


Figure 4. Detected and annotated elements visaliazation within VRML language

propagated into the ontology, serving as an improved knowledge base for further processing and decision steps. The last step consists in annotating the different geometries. Vertical elements of certain characteristics can be annotated directly. In more sophisticated cases, our prototype allows the combination of semantic information and topological ones that can deduce more robust results by minimizing the false acceptation rate. Finally, based on a list of SWRL rules, most of the detected geometries are annotated. In this example, among 13 elements are classified as Masts, 15 as Schaltanlage, three basic signals and finally, three secondary signals.

However, next to our experience, some limits are encountered. They are especially related very small elements detection and qualification where some noise on the ground still considered as semantic element. From our point of view, we think that the reason for such a false annotation is the lack of semantic characteristics related to such elements since until now; there is no real internal or external topology, neither internal geometric characteristic that discriminate such an element compared to others.

The created WiDOP platform offers the opportunity to materialize the annotation process by the generation and the visualization based on a VRML structure alimented from the knowledge base. It ensures an interactive visualization of the resulted annotation elements beginning from the initial state, to a set of intermediate states coming finally to an ending state, (see Figure 4), where the set of swrl rules are totally executed.

VI. CONCLUSION AND FUTURE WORKS

We have presented an automatic system for survey information model creation based on semantic knowledge modeling. Our solution aims to perform the detection of objects from laser scanner technology by using available knowledge about a specific domain (DB). The designed prototype as simple, as efficient and intelligent it is since it takes 3D point clouds of a facility and produce fully annotated scene within a VRML model file. The suggested

 ISBN: 978-1-61208-175-5

solution for this challenging problem has proven its efficiency through real tests within the Deutsche Bahn scene. The creation of processing and topological Built-Ins has presented a robust solution to resolve our problematic and to prove the ability of the semantic web language to intervene in any domain and create the difference.

Future work will include a more robust identification and annotation process of objects based on each object characteristics add to the integration of new 3D parameter knowledge's that can intervene within the detection and annotation process to make the process more flexible and intelligent.

ACKNOWLEDGMENT

This paper presents work performed in the framework of the research project funded by the German ministry of research and education under contract No. 1758X09. The authors cordially thank for this funding. Special thinks also for Hung Truong, Andreas Marbs, Ashish Karmacharya, Yoann Rous and Romain Tribotté for their contribution.

REFERENCES

[1] Wessel, R., Wahl, R., Klein, R., and Schnabel R, "Shape recognition in 3D point clouds," in Proc. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision., vol. 2, 2008.

[2] Kim, A., Funkhouser, VG., and Golovinskiy, T. "Shape-based recognition of 3d point clouds in urban environments," in IEEE 12th International Conference on Computer Vision, pp. 2154-2161, 2009.

[3] Cruz, C., Nicolle, C., and Duan, Y. "Architectural Reconstruction of 3D Building Objects through Semantic Knowledge Management," in 11th ACIS International Conference on Software Engineering, Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD), pp. 261-266, 2010.

[4] Ben Hmida,H., Cruz, C., Nicolle, C., and Boochs, F. "Semantic-based Technique for the Automation the 3D Reconstruction Process," in SEMAPRO 2010, The Fourth International Conference on Advances in Semantic Processing, Florence, Italy, pp. 191-198, 2010.

[5] VRML Virtual Reality Modeling Language. (1995, Apr.) W3C. [Online]. http://www.w3.org/MarkUp/VRML/. The last access date: 09-2011.

[6] Vanland, R., Nicolle, C., and Cruz, C. "IFC and building lifecycle management," Automation in Construction, vol. 18, pp. 70-78, 2008.

[7] Vosselman, S. "Extracting windows from terrestrial laser scanning," in Intl Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 36, pp. 12-14, 2007.

[8] Marton, R B., Blodow, Z C., Holzbach, N., Beetz, A., and Rusu, M. "Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments," in IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pp. 3601-3608, 2009.

[9] Ben Hmida, H., Marbs, A., Truong, H., Karmacharya, A., Cruz, C., Habed, A., Nicolle, C., Voisin,Y., and Boochs, F. "Integration of knowledge to support automatic object reconstruction from images and 3D data," in International Multi-Conference on Systems, Signals & Devices, Sousse Tunisia, March 22-25, 2011.

[10] Cantzler, H. "Improving architectural 3D reconstruction by constrained modelling," College of Science and Engineering. School of Informatics, 2003.

[11] Leica Cyclone. [Online]. http://hds.leica-geosystems.com/en/Leica-Cyclone_6515.htm. The last access date: 09-2011.

[12] Andreas, N., "Automatic Model Refinement for 3D Reconstruction with Mobile Robots," Fourth International Conference on 3-D Digital Imaging and Modeling, 3DIM, pp. 394-401, 2003.

[13] Goldberg, H.E. "State of the AEC industry: BIM implementation slow, but inevitable," Revista CADalyst, maio, 2005.

[14] Hajian, H. and Becerik-Gerber, B. "A Research Outlook for Real-Time Project Information Management by Integrating Advanced Field Data Acquisition Systems and Building Information Modeling," , 2009.

[15] Leadwerks Corporation. (2006) What is Constructive Solid Geometry?[Online]. http://www.leadwerks.com/files/csg.pdf. The last access date: 09-2011.

[16] OPEN CASCADE. (2000) OpenCascade - an open source library for BRep solid modeling. [Online]. http://www.opencascade.org/. The last access date: 09-2011.

[17] Campbell, R.J. and Flynn, P.J. "A survey of free-form object representation and recognition techniques," Computer Vision and Image Understanding, vol. 81, pp. 166-210, 2001.

[18] Pu, S. and Vosselman, G, "Knowledge based reconstruction of building models from terrestrial laser scanning data," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 64, pp. 575-584, 2009.

[19] Huber, K., Akinci, D., Krishnamurti, B., Yue, R. "The ASDMCon project: The challenge of detecting defects on construction sites," International Symposium on 3D Data Processing Visualization and Transmission, vol. 0, pp. 1048-1055, 2006.

[20] Haas, K., Bosche, CT. "Automated retrieval of 3D CAD model objects in construction range images," Automation in Construction, vol. 17, pp. 499-512, 2008.

[21] Marzani, F., Boochs, F., Cruz, C. "Ontology-driven 3D reconstruction of architectural objects," VISAPP (Special Sessions), pp. 47-54, 2007.

[22] Gruber, T R. "A translation approach to portable ontology specifications," Knowledge acquisition, vol. 5, pp. 199-220, 1993.

[23] Dickinson, J J., Dollin, I., Reynolds, C., Seaborne, D., Wilkinson, A., and Carroll, K. "Jena: implementing the semantic web recommendations," in Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, pp. 74-83, 2004.

[24] Patel-Schneider, I., Boley, P F., Tabet, H., Grosof, S., Dean, B., and Horrocks, M. "SWRL: A semantic web rule language combining OWL and RuleML," W3C Member submission, vol. 21, p. 79, 2004.

[25] Alles über Stellwerke. (2007) [Online]. www.stellwerke.de/. The last access date: 09-2011.

 ISBN: 978-1-61208-175-5

Light-weight Ontology Versioning with Multi-temporal RDF Schema

Fabio Grandi
Dipartimento di Elettronica, Informatica e Sistemistica
Alma Mater Studiorum – Università di Bologna
Bologna, Italy
Email: fabio.grandi@unibo.it

***Abstract*—In this paper, we present a multi-temporal RDF data model, which can be used to support RDF(S) light-weight ontology versioning. The data model is equipped with ontology change operations, which are defined in terms of low-level updates acting on RDF triples. As a result, the operational semantics of a complete set of primitive ontology changes has been formalized, taking care of preservation of class/property hierarchies and typing constraints. When used within the transaction template, which has also been introduced, the proposed ontology changes allow knowledge engineers or maintainers of semantics-based Web resources to easily define and manage temporal versions of an RDF(S) ontology.**

Keywords-ontology; versioning; temporal data; RDF(S).

I. INTRODUCTION

In some application domains, when an ontology is changed, the past version is required to be kept in addition to the new version (e.g., to maintain compatibility with applications and resources referencing the past one), giving rise to multi-version ontologies. Agents in such domains may often have to deal with a past perspective, like a Court having to judge today on a fact committed several years ago in the legal domain, where ontologies must evolve as a natural consequence of the dynamics involved in normative systems [9]. Moreover, several time dimensions are usually important for computer applications in such domains.

In this vein, we previously considered in [9] ontologies encoded in OWL/XML format and defined a temporal data model for the storage and management of multi-version ontologies in such a format. In [5], [6], we indeed considered ontologies serialized as RDF graphs [19], and introduced a multidimensional temporal data model and query language for the storage and management of multi-version ontologies in RDF format. In particular, since the triple store technology [20] for RDF is supposed to provide scalability for querying and retrieval, the temporal RDF data model we introduced in [5] is aimed at preserving the scalability property of such an approach as much as possible also in the temporal setting. This has been accomplished through the adoption of *temporal elements* [4], [12] as timestamps and a careful definition of the operational semantics of modification statements, which prevents the proliferation of *value-equivalent* triples even in the presence of multiple temporal dimensions.

In this work, we further focus on *light-weight ontologies* expressed with RDF(S), that is based on the vocabulary defined in RDF Schema [18], which are widespread and present a fast sharing rate in the loosely controlled and distributed environment of the Web and Web 2.0 [15]. Relying on the data in [3], Theoaris et al. estimate that 85.45% of the Semantic Web schemas are expressed in RDF(S) [14]. Hence, we will introduce in Sec. II a multi-temporal data model and an operation set, which can be used to support temporal versioning of RDF(S) ontologies. In particular, valid and transaction time dimensions and the types of ontology versioning, which stem from their adoption, are presented in Sec. II-A, the adopted underlying temporal RDF data model is briefly sketched in Sec. II-B, a comprehensive model for temporal RDF(S) ontology versioning is introduced in Sec. II-C and the definition of a complete set of ontology change primitives is provided in Sec. II-D. Conclusions will be finally found in Section III.

II. A MULTI-TEMPORAL RDF(S) DATA MODEL FOR LIGHT-WEIGHT ONTOLOGY VERSIONING

As RDF Schemas are quite similar to the type system of an object-oriented langauge and, thus, an ontology definition via RDFS closely resembles an object-oriented database schema, one could think to apply temporal schema versioning techniques like those in [8] to ontology versioning. However, there are two main differences between such two worlds, which make this application non straightforward. The first difference is that properties are first-class objects in RDFS and, thus, they cannot be dealt with as components of a class type, like in an object-oriented schema, but must be managed independently. The link between classes and properties is supplied by a sort of third-party tool, represented by domain and range definitions. The second difference is that, whereas in object-oriented databases we can separate the intensional (schema change) and the extensional (change propagation) aspects, in RDF(S) ontologies the two aspects are strictly related, since instances are part of the ontologies themselves and, thus, some ontology changes cannot be performed without affecting instances [16]. However, we will see in Sec. II-C that, with the proposed approach, both these aspects will turn into an advantage.

 ISBN: 978-1-61208-175-5

A. Multitemporal Ontology Versioning

In the temporal database literature, two time dimensions are usually considered: **valid time** (concerning the real world) and **transaction time** (concerning the computer life) [12]. With these time dimensions, likewise schema versioning in databases [2], [8], three kinds of temporal versioning can also be considered for ontology versioning:

- **Transaction-time ontology versioning** allows *on-time* ontology changes, that is ontology changes that are effective when applied. In this case, the management of time is completely transparent to the user: only the current ontology can be modified and ontology changes are effected in the usual way, without any reference to time. However, support of past ontology versions is granted by the system non-deletion policy, so that the user can always *rollback* the full ontology to a past state of its life.
- **Valid-time ontology versioning** is necessary when *retroactive* or *proactive* modifications [2] of an ontology have to be supported and it is useful to assign a temporal *validity* to ontology versions. With valid-time ontology versioning, multiple ontology versions, valid at different times, are all available for reasoning and for accessing and manipulating instances. The newly created ontology version can be assigned any validity by the designer, also in the past or future to effect retro- or pro-active ontology modifications, respectively. The (portions of) existing ontology versions overlapped by the validity of the new ontology version are overwritten.
- **Bitemporal ontology versioning** uses both time dimensions, that is retro- and pro-active ontology updates are supported in addition to transaction-time ontology versioning. With respect to valid-time ontology versioning, the complete history of ontology changes is maintained as no ontology version is ever discarded (overlapped portions are "archived" rather than deleted). In a system where full auditing/traceability of the maintenance process is required, only bitemporal ontology versioning allows verifying whether an ontology version was created by a retro- or pro-active ontology change.

Other time dimensions can also be considered as further (orthogonal) versioning dimensions [5] in special application domains, like *efficacy* or *applicability* time in the legal or medical domains [7], [9].

B. A multi-temporal RDF database model

We briefly recall here the base definitions of the multi-temporal RDF database model [5] underlying our proposal, starting from an N-dimensional time domain:

$$\mathcal{T} = \mathcal{T}_1 \times \mathcal{T}_2 \times \cdots \times \mathcal{T}_N$$

where $\mathcal{T}_i = [0, \mathrm{UC}]_i$ is the i-th time domain. Right-unlimited time intervals are expressed as $[t, \mathrm{UC}]$, where UC means "Until Changed", though such a symbol is often used in temporal database literature [12] for transaction time only (whereas, e.g., "forever" or ∞ is used for valid time). Such naming choice refers to the modeling of time-varying data, which are potentially subject to change with respect to all the considered time dimensions. Then, a multi-temporal RDF triple is defined as $(s, p, o \,|\, T)$, where s is a subject, p is a property, o is an object and $T \subseteq \mathcal{T}$ is a *timestamp* assigning a *temporal pertinence* to the RDF triple (s, p, o). We will also call the (non-temporal) triple (s, p, o) the value or the contents of the temporal triple $(s, p, o \,|\, T)$. The temporal pertinence of a triple is a subset of the multidimensional time domain, which is represented by a *temporal element* [12], that is a disjoint union of multidimensional temporal intervals, each one obtained as the Cartesian product of one time interval for each of the supported temporal dimensions. A multi-temporal RDF database is defined as a set of timestamped RDF triples:

$$\text{RDF-TDB} = \{\, (s, p, o \,|\, T) \mid T \subseteq \mathcal{T} \,\}$$

with the integrity constraint:

$$\forall (s, p, o \,|\, T), (s', p', o' \,|\, T') \in \text{RDF-TDB}:$$
$$s = s' \wedge p = p' \wedge o = o' \Longrightarrow T = T'$$

which requires that no value-equivalent distinct triples exist. The adoption of timestamps made-up of temporal elements instead of (multi-temporal) simple intervals avoids the duplication of triples in the presence of a temporal pertinence with a complex shape [5].

In practice, we store different triple versions only once with a complex timestamp rather than storing multiple copies of them with a simple timestamp as in other RDF temporal extensions [10], [17], [23]. The memory saving we obtain grows with the dimensionality of the time domain but could be relevant also with a single time dimension [5]. Moreover, since temporal elements are closed under set union, intersection and complementation operations, they lead to query languages that are more natural [4].

The data model is equipped with three modification operations —`INSERT`, `DELETE` and `UPDATE`— with a SQL-like syntax also inspired by the SPARQL/Update proposal [22], and whose semantics has been defined in [5] in such a way, that the integrity constraints concerning temporal elements and the controlled growth of value-equivalent triples made possible by temporal elements are automatically maintained. This fact ensures that, in a temporal setting and compatibly with the growth of the number of versions, unlike other approaches, the scalability property of the triple storage technology is preserved.

C. Temporal Versioning of Light-weight RDF(S) Ontologies

The signature of a Light-weight RDF(S) Ontology [15] can be defined as follows:

$$\mathcal{O} = (\mathcal{C}, \mathcal{H}_\mathcal{C}, \mathcal{I}_\mathcal{C}, \mathcal{P}, \mathcal{H}_\mathcal{P}, \mathcal{I}_\mathcal{P}, \mathcal{D}_\mathcal{P}, \mathcal{R}_\mathcal{P})$$

 ISBN: 978-1-61208-175-5

where $\mathcal{C}$ is the set of classes, $\mathcal{H}_\mathcal{C}$ is the class hierarchy, and $\mathcal{I}_\mathcal{C}$ is the set of class instances; $\mathcal{P}$ is the set of properties, $\mathcal{H}_\mathcal{P}$ is the property hierarchy, and $\mathcal{I}_\mathcal{P}$ is the set of property instances; finally $\mathcal{D}_\mathcal{P}$ and $\mathcal{R}_\mathcal{P}$ are, respectively, the set of property domain and range definitions. Hence, a *Multi-temporal Light-weight RDF(S) Ontology* can be defined as a multi-temporal RDF database as:

$$
\begin{aligned}
&O := \\
&\{(C, \texttt{rdf:type}, \texttt{rdfs:Class} \,|\, T) \,|\, C \in \mathcal{C}, T \subseteq \mathcal{T}\} \cup \\
&\{(C, \texttt{rdfs:subClassOf}, C' \,|\, T) \,|\, (C, C') \in \mathcal{H}_\mathcal{C}, T \subseteq \mathcal{T}\} \cup \\
&\{(x, \texttt{rdf:type}, C \,|\, T) \,|\, C(x) \in \mathcal{I}_\mathcal{C}, T \subseteq \mathcal{T}\} \cup \\
&\{(P, \texttt{rdf:type}, \texttt{rdf:Property} \,|\, T) \,|\, P \in \mathcal{P}, T \subseteq \mathcal{T}\} \cup \\
&\{(P, \texttt{rdfs:subPropertyOf}, P' \,|\, T) \,|\, (P, P') \in \mathcal{H}_\mathcal{P}, T \subseteq \mathcal{T}\} \cup \\
&\{(x, P, y \,|\, T) \,|\, P(x, y) \in \mathcal{I}_\mathcal{P}, T \subseteq \mathcal{T}\} \cup \\
&\{(P, \texttt{rdfs:domain}, C \,|\, T) \,|\, \mathrm{dom}(P, C) \in \mathcal{D}_\mathcal{P}, T \subseteq \mathcal{T}\} \cup \\
&\{(P, \texttt{rdfs:range}, C \,|\, T) \,|\, \mathrm{range}(P, C) \in \mathcal{R}_\mathcal{P}, T \subseteq \mathcal{T}\}
\end{aligned}
$$

This definition is useful to understand how ontology changes will be mapped onto manipulation of temporal triples. The general template, which can be used for a transaction which creates a new ontology version, is exemplified in Fig. 1. Such a transaction needs two temporal parameters as inputs: the *ontology selection validity* and the *ontology change validity* (corresponding to schema selection validity and schema change validity in databases [8]). The former (*OS_Validity*) is a valid-time point and is used to select the ontology version —not necessarily the present one— chosen as the starting base, to which ontology changes are applied to produce the new version; the latter (*OC_Validity*) is a valid-time element, that is a disjoint union of valid-time intervals, representing the validity to be assigned to new version resulting form the application of the ontology changes. As far as transaction time is concerned, default conditions are used, since only current ontology versions can be chosen as modification base and the new version must be assigned a [NOW,UC] pertinence.

In Fig. 1, statements 1 and 7 are SQL-like syntactic delimiters for the transaction body. As a first operation (2), a temporary (non-temporal) RDF graph is created to be used as work version of the ontology. Such graph is then populated (3) with the RDF triples making up the work version, extracted as a *snapshot query* from the temporal ontology (i.e., the triples whose temporal pertinence contains *OS_Validity*$\times\{$NOW$\}$, with the timestamp projected out). Then, the required sequence of (non-temporal) ontology changes is applied to the work version. When the new ontology version is ready, it must be loaded into the temporal ontology with the desired time pertinence *OC_Validity*$\times$[NOW,UC]. To this end, the contents of the temporal ontology within the time window *OC_Validity*$\times$[NOW,UC] are deleted (4), in order to make room for the new version in the time domain, and the triples making up the work version are inserted as temporal triples into the temporal ontology with the assigned timestamp *OC_Validity*$\times$[NOW,UC] (5). After that, the temporary work version is no longer necessary and can be discarded (6). Notice that, whereas statements 2 and 6 are "standard" (i.e., non-temporal) SPARQL/Update instructions [22], statements 3, 4 and 5 are temporal T-SPARQL operations as defined in [5], [6].

Adopting the template in Fig. 1, schema changes are applied on the work version, which is a traditional (non-temporal) RDF(S) ontology and, thus, there is no need to introduce *temporal* schema change operations. Hence, as a set of possible schema changes, we could even consider the operations made available by an existing ontology editor [21]. Differently from other approaches with a strong logic-based ground (e.g., [11], [13]), our choice is to follow the simpler approach previously used for database schema versioning (e.g., [1], [8]) by considering the set of schema changes in Tab. I: the proposed operations are *primitive*, as each of them acts on a single element of an RDFS ontology and none of them can be decomposed in terms of the others, and make up a complete set. Completeness can easily be checked by verifying that any arbitrarily complex RDFS ontology can be built from scratch (or completely destroyed) via the execution of a suitable sequence of ontology change primitives.

With this approach, we remit in fact the management of the resulting ontology version validity to the responsibility of the designer. For instance, the validity rule R8 enforced by the approach in [13], stating that "the domain of a property is unique", would be too limiting and, thus, unacceptable with respect to the requisites of several application domains. Notice that, in the formalization of some real world component, which is fruit of some human (i.e., arbitrary or at least non completely rational) activity, like the legal domain, a correctly designed ontology could even be logically inconsistent. For example, in several countries, the primary function of the Supreme/Constitutional Court is to rule conflicts between ordinary norms and constitutional laws. As long as such conflicts exist, the whole corpus of regulations is in fact logically inconsistent and as such must be modelled.

The semantics of the primitives in Tab. I will be defined in the next section, taking care of preservation of class/property hierarchies and typing constraints (like in an object-oriented database schema). Moreover, since instances are part of the ontology definition, we do not have in this framework to deal with the interaction between versioning at intensional and extensional levels, extensively discussed in [8, Sec. 4], arising in databases where schemata and data are possibly versioned along different time dimensions.

We underline that, whereas the proposed operation set could also be used for ontology evolution in a non-temporal setting, where only the current version of the ontology is retained, their usage within the template in Fig. 1 gives rise to full-fledged temporal ontology versioning.

 ISBN: 978-1-61208-175-5

```
1. BEGIN TRANSACTION ;
2. CREATE GRAPH <http://example.org/workVersion> ;
3. INSERT INTO <http://example.org/workVersion> { ?s ?p ?o }
   WHERE { TGRAPH <http://example.org/tOntology> { ?s ?p ?o | ?t } .
           FILTER(VALID(?t) CONTAINS OS_Validity && TRANSACTION(?t) CONTAINS fn:current-date()) } ;

   ⇒ a sequence of ontology changes acting on the (non-temporal) workVersion graph goes here

4. DELETE FROM <http://example.org/tOntology> { ?s ?p ?o } VALID OC_Validity ;
5. INSERT INTO <http://example.org/tOntology> { ?s ?p ?o } VALID OC_Validity
   WHERE { GRAPH <http://example.org/workVersion> { ?s ?p ?o } } ;
6. DROP GRAPH <http://example.org/workVersion> ;
7. COMMIT TRANSACTION
```

Figure 1. Template for a transaction implementing the derivation of a new ontology version.

Changes to the class collection
CREATE_CLASS(*NewClass*)
DROP_CLASS(*Class*)
RENAME_CLASS(*Class, NewName*)
Changes to the property collection
CREATE_PROPERTY(*NewProperty*)
DROP_PROPERTY(*Property*)
RENAME_PROPERTY(*Property, NewName*)
Changes to the class and property hierarchies
ADD_SUBCLASS(*SubClass, Class*)
DELETE_SUBCLASS(*SubClass, Class*)
ADD_SUBPROPERTY(*SubProperty, Property*)
DELETE_SUBPROPERTY(*SubProperty, Property*)
Changes to the property domain and range definitions
ADD_DOMAIN(*Property, NewDomain*)
ADD_RANGE(*Property, NewRange*)
DELETE_DOMAIN(*Property, Domain*)
DELETE_RANGE(*Property, Range*)
CHANGE_DOMAIN(*Property, Domain, NewDomain*)
CHANGE_RANGE(*Property, Range, NewRange*)

Table I
LIST OF PRIMITIVE RDFS ONTOLOGY CHANGES.

D. Definition of RDF(S) Ontology Changes

In this section, we show how the primitive ontology change operations in Tab. I can be defined in terms of manipulation operations acting on the RFD(S) contents of the work version. An SQL-like syntax (which seems a bit more intuitive than SPARQL/Update [22] for SQL-acquainted readers) is used to express INSERT, DELETE and UPDATE statements acting on RDF triples. In the following definitions, for the sake of simplicity, although non explicitly written all the manipulation operations are supposed to work on the named graph <http://example.org/workVersion> when embedded into the transaction template of Fig. 1.

The CREATE_CLASS primitive adds a new class to the set of classes $\mathcal{C}$ and can simply be defined as:

```
CREATE_CLASS(NewClass) :=
   INSERT { NewClass rdf:type rdfs:Class }
```

The DROP_CLASS primitive eliminates a class from the ontology. This means that the class must be removed from the set $\mathcal{C}$ and from the class hierarchy $\mathcal{H}_C$, all the domain and range definitions having the class as target must be removed from $\mathcal{D}_P$ and $\mathcal{R}_P$, respectively, and all the instances of the class must also be removed from $\mathcal{I}_C$. Thus, it can be defined as follows:

```
DROP_CLASS(Class) :=
   DELETE { Class rdf:type rdfs:Class } ;
   INSERT { ?C rdfs:subClassOf ?D } ;
      WHERE { ?C rdfs:subClassOf Class .
              Class rdfs:subClassOf ?D } ;
   DELETE { Class rdfs:subClassOf ?C } ;
   DELETE { ?C rdfs:subClassOf Class } ;
   DELETE { ?P rdfs:domain Class } ;
   DELETE { ?P rdfs:range Class } ;
   DELETE { ?X rdf:type Class }
```

Notice that, before Class can be removed from $\mathcal{H}_C$, if $\{(C, \texttt{Class}), (\texttt{Class}, D)\} \subseteq \mathcal{H}_C$, an explicit inheritance link (C, D) must be added to $\mathcal{H}_C$ in order to maintain the continuity of the inheritance hierarchy. We assume the relation rdfs:subClassOf is not interpreted here as transitive (i.e., it only matches explicitly stored triples), so that only one explicit link is added. In this way, we also produce a consistent state without explicitly computing the transitive closure of the inheritance relation, which would increase the number of stored triples in the work version.

The RENAME_CLASS primitive changes the name of a class in the ontology. This means that the name must be changed wherever the class occurs, that is in $\mathcal{C}$, $\mathcal{H}_C$, $\mathcal{D}_P$, $\mathcal{R}_P$ and $\mathcal{I}_C$. The primitive can be defined as follows:

```
RENAME_CLASS(Class, NewName) :=
   UPDATE { Class rdf:type rdfs:Class }
      SET { NewName rdf:type rdfs:Class } ;
   UPDATE { Class rdfs:subClassOf ?C }
      SET { NewName rdfs:subClassOf ?C } ;
   UPDATE { ?C rdfs:subClassOf Class }
      SET { ?C rdfs:subClassOf NewName } ;
   UPDATE { ?P rdfs:domain Class }
      SET { ?P rdf:domain NewName } ;
   UPDATE { ?P rdfs:range Class }
      SET { ?P rdf:range NewName } ;
   UPDATE { ?X rdf:type Class }
```

 ISBN: 978-1-61208-175-5

```
    SET { ?X rdf:type NewName } ;
```

Obviously, the execution of

```
RENAME_CLASS(ex:C,ex:D)
```

is not equivalent to the sequence:

```
DELETE_CLASS(ex:C) ;
CREATE_CLASS(ex:D)
```

because, in the former case, the instances of class `ex:C` are preserved and assigned to `ex:D` (and also instances of properties having `ex:C` as domain or range are preserved), whereas, in the latter, instances are discarded.

The `CREATE_PROPERTY` primitive adds a new class to the set of properties $\mathcal{P}$ and can simply be defined as:

```
CREATE_PROPERTY(NewProperty) :=
   INSERT { NewProperty rdf:type rdf:Property }
```

The `DROP_PROPERTY` primitive eliminates a property from the ontology. This means that the property must be removed from the set $\mathcal{P}$ and from the property hierarchy $\mathcal{H}_P$, all the domain and range definitions having the property as source must be removed from $\mathcal{D}_P$ and $\mathcal{R}_P$, respectively, and all the instances of the property must also be removed from $\mathcal{I}_P$. Thus, it can be defined as follows:

```
DROP_PROPERTY(Property) :=
   DELETE { Property rdf:type rdf:Property } ;
   INSERT { ?P rdfs:subPropertyOf ?Q }
      WHERE { ?P rdfs:subPropertyOf Property .
              Property rdfs:subPropertyOf ?Q } ;
   DELETE { Property rdfs:subPropertyOf ?P } ;
   DELETE { ?P rdfs:subPropertyOf Property } ;
   DELETE { Property rdfs:domain ?C } ;
   DELETE { Property rdfs:range ?C } ;
   DELETE { ?X Property ?Y }
```

As for classes, the deletion of the property in the middle of a inheritance path requires the insertion of a new explicit inheritance link to $\mathcal{H}_P$ before the property is removed, in order not to break the path into two stumps (we also assume the relation `rdfs:subPropertyOf` is not interpreted here as transitive, so that only one explicit link is added).

The `RENAME_PROPERTY` primitive changes the name of a property in the ontology. This means that the name must be changed wherever the property occurs, that is in $\mathcal{P}$, $\mathcal{H}_P$, $\mathcal{D}_P$, $\mathcal{R}_P$ and $\mathcal{I}_P$. The primitive can be defined as follows:

```
RENAME_PROPERTY(Property, NewName) :=
   UPDATE { Property rdf:type rdf:Property }
      SET { NewName rdf:type rdf:Property } ;
   UPDATE { Property rdfs:subPropertyOf ?P }
      SET { NewName rdfs:subPropertyOf ?P } ;
   UPDATE { ?P rdfs:subPropertyOf Property }
      SET { ?P rdfs:subPropertyOf NewName } ;
   UPDATE { Property rdfs:domain ?D }
      SET { NewName rdfs:domain ?D } ;
   UPDATE { Property rdfs:range ?D }
      SET { NewName rdfs:range ?D } ;
   UPDATE { ?X Property ?Y }
      SET { ?X NewName ?Y }
```

The `ADD_SUBCLASS` primitive is used to add a new inheritance link to the class hierarchy $\mathcal{H}_C$ and can simply be defined as:

```
ADD_SUBCLASS(SubClass, Class) :=
   INSERT { SubClass rdfs:subClassOf Class }
```

The `DELETE_SUBCLASS` primitive is used to remove an inheritance link from the class hierarchy $\mathcal{H}_C$ and can simply be defined as:

```
DELETE_SUBCLASS(SubClass, Class) :=
   DELETE { SubClass rdfs:subClassOf Class }
```

The `ADD_SUBPROPERTY` primitive is used to add a new inheritance link to the property hierarchy $\mathcal{H}_P$ and can simply be defined as:

```
ADD_SUBPROPERTY(SubProperty, Property) :=
   INSERT
      { SubProperty rdfs:subPropertyOf Property }
```

The `DELETE_SUBPROPERTY` primitive is used to remove an inheritance link from the property hierarchy $\mathcal{H}_P$ and can simply be defined as:

```
DELETE_SUBPROPERTY(SubProperty, Property) :=
   DELETE
      { SubProperty rdfs:subPropertyOf Property }
```

The `ADD_DOMAIN` primitive is used to add a new domain relationship to $\mathcal{D}_P$ and can be defined as:

```
ADD_DOMAIN(Property, NewDomain) :=
   INSERT { Property rdfs:domain NewDomain } ;
   INSERT { ?X rdf:type NewDomain }
      WHERE { ?X Property ?Y }
```

Notice that, in accordance to [18], properties are allowed to have multiple domains and the resources denoted by subjects of triples with predicate *Property* must be instances of all the classes stated by the `rdfs:domain` properties. Hence, a new instance *NewDomain*(x) must be added to $\mathcal{I}_C$ for each instance *Property*$(x, y) \in \mathcal{I}_P$.

The `ADD_RANGE` primitive is used to add a new range relationship to $\mathcal{R}_P$ and can be defined as:

```
ADD_RANGE(Property, NewRange) :=
   INSERT { Property rdfs:range NewRange } ;
   INSERT { ?Y rdf:type NewRange }
      WHERE { ?X Property ?Y }
```

Notice that, in accordance to [18], properties are allowed to have multiple ranges and the resources denoted by objects of triples with predicate *Property* must be instances of all the classes stated by the `rdfs:range` properties. Hence, a new instance *NewRange*(y) must be added to $\mathcal{I}_C$ for each instance *Property*$(x, y) \in \mathcal{I}_P$.

 ISBN: 978-1-61208-175-5

The DELETE_DOMAIN primitive is used to remove a domain relationship of a property. This means that the domain must be removed from $\mathcal{D}_P$ together with all the instances of the property referencing the domain, which must be removed from $\mathcal{I}_P$. The operation can then be defined as:

```
DELETE_DOMAIN(Property, Domain) :=
   DELETE { Property rdfs:domain Domain } ;
   DELETE { ?X Property ?Y }
      WHERE { { ?X rdf:type Domain }
              UNION
              { ?C rdfs:subClassOf Domain .
                ?X rdf:type ?C }
            }
```

In this case, we assume the relation rdfs:subClassOf is interpreted as transitive during the evaluation of the statement, as we must delete all the instances *Property*$(x, y) \in \mathcal{I}_P$, where x is a member of *Domain* or of any of its subclasses along the inheritance hierarchy.

Similarly, the DELETE_RANGE primitive is used to remove a range relationship of a property. This means that the range must be removed from $\mathcal{R}_P$ together with all the instances of the property referencing the range, which must be removed from $\mathcal{I}_P$. The operation can be defined as:

```
DELETE_RANGE(Property, Range) :=
   DELETE { Property rdfs:range Range } ;
   DELETE { ?X Property ?Y }
      WHERE { { ?Y rdf:type Range }
              UNION
              { ?C rdfs:subClassOf Range .
                ?Y rdf:type ?C }
            }
```

Also in this case, we assume the relation rdfs:subClassOf is interpreted as transitive, as we must delete all the instances *Property*$(x, y) \in \mathcal{I}_P$, where y is a member of *Range* or of any of its subclasses along the inheritance hierarchy.

The CHANGE_DOMAIN primitive is used to change a property domain definition in $\mathcal{D}_P$ and can be defined as:

```
CHANGE_DOMAIN(Property, Domain, NewDomain) :=
   UPDATE { Property rdfs:domain Domain }
      SET { Property rdfs:domain NewDomain }
```

Analogously, the CHANGE_RANGE primitive to be used to change a property range definition in $\mathcal{R}_P$ can be defined as:

```
CHANGE_RANGE(Property, Range, NewRange) :=
   UPDATE { Property rdfs:range Range }
      SET { Property rdfs:range NewRange }
```

In the last two definitions, we assumed instances of Property are not affected by the domain or range changes. If this is not the case, suitable conversion functions must be supplied, as defined in a given namespace, to correctly propagate the change to instances (e.g., cfn:DomainToNewDomain and cfn:RangeToNewRange for literal data). For instance, in the case of CHANGE_RANGE, this can be done as follows:

```
PREFIX cfn: <http://example.org/conv_funct#>
UPDATE { ?X Property ?Y }
   SET { ?X Property cfn:RangeToNewRange(?Y) }
   WHERE { { ?Y rdf:type Range .
             FILTER(isLiteral(?Y) &&
             cfn:RangeToNewRange(?Y))!="") }
           UNION
           { ?C rdfs:subClassOf Range .
             ?Y rdf:type ?C .
             FILTER(isLiteral(?Y) &&
             cfn:RangeToNewRange(?Y))!="") }
         } ;
DELETE { ?X Property ?Y }
   WHERE { { ?Y rdf:type Range .
             FILTER(isLiteral(?Y) &&
             cfn:RangeToNewRange(?Y))="") }
           UNION
           { ?C rdfs:subClassOf Range .
             ?Y rdf:type ?C .
             FILTER(isLiteral(?Y) &&
             cfn:RangeToNewRange(?Y))="") }
         }
```

If the conversion function is able to produce a significant value (i.e., a non-empty string), the new value is used to update the property instances, also involving range subclasses. Otherwise, the property instances, which cannot be converted, are discarded. This correspond, in the terminology of schema evolution, to a combined deployment of the *coercion* and *filtering* techniques [8]. Notice that, for instance, the execution of:

```
CHANGE_DOMAIN(ex:P,ex:C,ex:D)
```

is not equivalent to the sequence:

```
DELETE_DOMAIN(ex:P,ex:C) ;
ADD_DOMAIN(ex:P,ex:D)
```

because, in the former case, the instances of property ex:P are preserved, if domains ex:C and ex:D are compatible or a conversion function exists, whereas, in the latter, instances are in any case discarded.

III. Conclusion and Future Works

In this work, we added another piece to our proposal, already including [5], [6], [9], which involves the extension to the Semantic Web of temporal data models and query languages developed in decades of temporal database research, by focusing on temporal versioning of light-weight ontologies expressed in RDF(S). To this end, we showed how the multi-temporal RDF data model [5] can easily be used to support RDF(S) ontology versioning. The data model has been equipped with a complete set of primitive ontology change operations, defined in terms of low-level modifications acting on RDF triples. Sequences of such ontology changes can simply be embedded into the transaction

 ISBN: 978-1-61208-175-5

template we proposed, to be used by knowledge engineers or maintainers of semantics-based Web resources in order to support full-fledged temporal ontology versioning.

In future research, we will consider the design and prototyping of a query engine supporting the execution of T-SPARQL manipulation operations, which implement the ontology change primitives described in this paper. We will also consider the adoption of suitable multidimensional index and storage structures to efficiently support temporal versioning of light-weight ontologies expressed in RDF(S).

REFERENCES

[1] J. Banerjee, W. Kim, H.-J. Kim, and H. F. Korth. Semantics and Implementation of Schema Evolution in Object-Oriented Databases. In *Proc. of SIGMOD Conference*, ACM Press, 1987, pp. 311–322.

[2] C. De Castro, F. Grandi, and M. R. Scalas. Schema Versioning for Multitemporal Relational Databases. *Information Systems*, vol. 22:5, 1997, pp. 249–290.

[3] L. Ding and T. W. Finin. Characterizing the Semantic Web on the Web. In *Proc. of ISWC Conference*, Springer-Verlag, LNCS No. 4273, 2006, pp. 242–257.

[4] S. Gadia. A Homogeneous Relational Model and Query Language for Temporal Databases. *ACM Transactions on Database Systems*, vol. 13:3, 1998, pp. 418–448.

[5] F. Grandi. Multi-temporal RDF Ontology Versioning. In *Proc. of IWOD Workshop*, CEUR-WS, 2009.

[6] F. Grandi. T-SPARQL: a TSQL2-like Temporal Query Language for RDF. In *Proc. of GraphQ Workshop*, CEUR-WS, 2010, pp. 21–30.

[7] F. Grandi. A Personalization Environment for Multi-Version Clinical Guidelines. In A. Fred, J. Filipe, and H. Gamboa, editors, *Biomedical Engineering Systems and Technologies 2010*, Springer-Verlag, CCIS No. 127, 2011, pp. 57–69.

[8] F. Grandi and F. Mandreoli. A Formal Model for Temporal Schema Versioning in Object-Oriented Databases. *Data & Knowledge Engineering*, vol. 46:2, 2003, pp. 123–167.

[9] F. Grandi and M. R. Scalas. The Valid Ontology: A Simple OWL Temporal Versioning Framework. In *Proc. of SEMAPRO Conference*, IEEE Computer Society, 2009, pp. 98–102.

[10] C. Gutierrez, C. A. Hurtado and A. A. Vaisman. Introducing Time into RDF. *IEEE Transactions on Knowledge and Data Engineering*, vol. 19:2, 2007, pp. 207–218.

[11] C. Gutierrez, C. A. Hurtado, and A. A. Vaisman. RDFS Update: From Theory to Practice. In *Proc. of ESWC Conference*, Springer-Verlag, LNCS No. 6644, 2011, pp. 93–107.

[12] C. S. Jensen, C. E. Dyreson (eds.), et al. The Consensus Glossary of Temporal Database Concepts - February 1998 version. In O. Etzion, S. Jajodia, and S. Sripada, editors, *Temporal Databases — Research and Practice*, Springer-Verlag, LNCS No. 1399, 1998, pp. 367–405.

[13] G. Konstantinidis, G. Flouris, G. Antoniou, and V. Christophides. A Formal Approach for RDF/S Ontology Evolution. In *Proc. of ECAI Conference*, IOS Press, 2008, pp. 70–74.

[14] Y. Theoharis, Y. Tzitzikas, D. Kotzinos, and V. Christophides. On Graph Features of Semantic Web Schemas. *IEEE Transactions on Knowledge and Data Engineering*, vol. 20:5, 2008, pp. 692–702.

[15] P. Mika and H. Akkermans. Towards a New Synthesis of Ontology Technology and Knowledge Management. *Knowledge Engineering Review*, vol. 19:4, 2004, pp. 317–345.

[16] N. F. Noy and M. C. A. Klein. Ontology Evolution: Not the Same as Schema Evolution. *Knowledge and Information Systems*, vol. 6:4, 2003, pp. 428–440.

[17] A. Pugliese, O. Udrea, and V. S. Subrahmanian. Scaling RDF with Time. In *Proc. of WWW Conference*, ACM Press, 2008, pp. 605–614.

[18] RDF Vocabulary Description Language 1.0: RDF Schema. W3C Consortium, http://www.w3.org/TR/rdf-schema/ [retrieved 2011-09-10].

[19] Resource Description Framework. W3C Consortium, http://www.w3.org/RDF/ [retrieved 2011-09-10].

[20] K. Rohloff, M. Dean, I. Emmons, D. Ryder and J. Summer. An Evaluation of Triple-store Technologies for Large Data Stores. In *Proc. of OTM Workshops*, Springer-Verlag, LNCS No. 4806, 2007, pp. 1105–1147.

[21] Semantic Web Tools. W3C Consortium, http://www.w3.org/2001/sw/wiki/SemanticWebTools [retrieved 2011-09-10].

[22] SPARQL Update. W3C Consortium, http://www.w3.org/Submission/SPARQL-Update/ [retrieved 2011-09-10].

[23] J. Tappolet, and A. Bernstein. Applied temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL. In *Proc. of ESWC Conference*, Springer-Verlag, LNCS No. 5554, 2009, pp. 302–322.

 ISBN: 978-1-61208-175-5

Local Theme Detection and Annotation with Keywords for Narrow and Wide Domain Short Text Collections

Svetlana V. Popova
Saint-Petersburg State University
Faculty of applied mathematics and control processing
Saint-Petersburg, Russia
spbu@bk.ru

Ivan A. Khodyrev
Saint-Petersburg State Electro-technical University
Faculty of computer science and informatics
Saint-Petersburg, Russia
kivan.mih @gmail.com

***Abstract*—This paper presents a clustering approach for text collections and automatic detection of topic and keywords for clusters. Present research focuses on narrow domain short texts such as short news and scientific paper abstracts. We propose a term selection method, which helps to significantly improve hierarchic clustering quality, and also the automatic algorithm to annotate clusters with keywords and topic names. The results of clustering are good comparing with the results of other approaches and our algorithm also allows extracting keywords for each cluster, using the information about the size of a cluster and word frequencies in documents.**

Keywords-narrow domain short text clustering; automatic annotation; hierarchical clustering; Pearson correlation.

I. INTRODUCTION

In the presented paper, we are solving two main tasks: clustering and annotation tasks with keywords for small collections of short texts. We have chosen two types of collections for our tasks: first type collections contain texts from one narrow domain and second type collections contain texts from different domains. In our experiments, we are using collections, which are used for clustering in other papers [2][8][9][12]. We also observe that there is not much attention paid in literature in respect to annotation of narrow domain short texts for small collections.

Topics/trends detection and annotation is a popular theme today. Annotations help user to understand if a document or a group of documents is useful in respect to his goals or not without reading the full source. Annotations also help in a search process when user tries to find documents similar to some target document. New keywords appearance in sets of scientific articles could signify emerge of a new research domain or a new trend in present domains. The task of novelty detection is highly demanded today, but it is also a hard task to deal with. Main themes detection in news collections is related to topic detection and tracking domain (TDT) [4][5][15]. Keyword detection and annotation for document collections could be used in automated ontology's creation task.

The task of short text processing and analysis is emerged with the development of social networks. Today, the practical interest to analyze messages in blogs, forums, e-mails, sms is constantly growing [3][16]. There is a wide variation of tasks in this field: social analysis, opinion mining and sentiment analysis, searching for useful and redundant branches on forums, social network search engines etc. Electronic libraries also benefit from the research in the field of short texts, because it could help automating searching and sorting documents by using abstracts.

The importance to separate small collections could be defined as follows. Consider an analysis of text documents' collection with clustering goal. It leads to situation where from big collections small subgroups of texts are extracted, which need further processing. Analysis of these subgroups needs changes in text processing. Small sizes of texts and collections which contain them make word evaluation a hard task, because amount of data is very limited

We are basing annotation results of preceding clustering. So our first task was clustering. Short texts clustering is a task with high complexity [2][8][12]. In present paper, we propose clustering approach based on Pearson correlation coefficient [19] and special term selection technique.

As a clustering algorithm we are using one of the hierarchical clustering algorithms [7][18] and Pearson correlation as measure between texts. On term selection step not more than 10% of a collection's vocabulary left. Our research showed that quality of clustering is increased if words with high value of document frequency are used, with exception to some words with the highest document frequency. Obtained clustering results are relatively good comparing with the other methods [2][8][12]. Approach based on Pearson correlation measure seems productive and we are planning to test it with different clustering algorithms in the future. There is still unsolved question: how to determine the right number of clusters for hierarchical clustering algorithm.

Second task is annotation of given type of collections. In this paper, we consider only keyword annotation. Word's overlapping between clusters makes this task difficult. Choosing frequent words in some cluster as a keyword usually lead to situation where common word for the whole

 ISBN: 978-1-61208-175-5

collection is choosing which is not informative for cluster. From the other hand, setting a threshold for a words which appear outside of cluster, could lead to loss of semantically significant words. In present paper we propose novel algorithm which helps to deal with these problems.

The rest of the paper organized as follows. In section 2, we describe related work. In section 3, we present test collections and the measure, with which we could compare the results automatic and manual clustering. In section 4, proposed clustering algorithm, term selection method and keywords detection algorithm are described. Section 5 contains experimental results, and we make a conclusion in section 6.

II. RELATED WORK

Clustering of narrow domain short text collections was addressed in David Pinto's PhD and in [12]. Pinto tested a number of algorithms, similarity measures to compare documents and term selection techniques. Pinto suggests that it is possible to increase the clustering quality using self-term expansion before term selection. Idea of self-term expansion was further developed in [13]. In [11], weblog clustering task is solving using different topics detection inside documents with preceding self-term expansion. The best clustering results for narrow domain short texts were obtained in [2][8][9]. In [2], algorithm CLUDIPSO is introduced; it is based on discrete particle swarm optimization. It needs precise information about the number of clusters and some other parameters, which were calculated in [2] during experiments. However even for fixed parameters on the same date, the quality of CLUDIPSO's clustering result could vary. In [8], Ant-Tree-Silhouette-Attraction algorithm (*AntSA*) was introduced, which is based on AntTree algorithm and use some initial data partitions by using CLUDIPSO (*AntSA-CLU*). *AntSA-CLU* gives better results comparing to CLUDIPSO, but it also needs input parameters to be set and the result may vary from experiment to experiment as well. In [9], iterative method for short text clustering tasks (ITSA) was proposed. This method does not make clustering itself, but it integrates and refines results of arbitrary clustering algorithms and based on them generates final result.

In [2][8][9][12], authors show clustering results on narrow domain short texts using different algorithms: Single Link Clustering, Complete Link Clustering, K-Nearest Neighbour, K-Star and a modified version of the K-Star method (NN1), K-means, MajorClust, CHAMELEON, DBSCAN. Obtained results are relatively low for these algorithms. Algorithms which show the best results (CLUDIPSO, *AntSA-CLU*) do not show these results constantly on narrow domain collections with low topics differentiation. Clustering quality changes on each independent run for these algorithms and it could vary: it could be very good or it could be relatively low on different runs on the same data with the same input parameters. In practice such situation is usually does not satisfy user because when user receives bad results from some algorithm a number of times, he will most likely stop using it. So for presented work we have chosen hierarchical clustering algorithms, which give the same result for fixed number of clusters. We defined the term selection method and similarity measure between documents to reach results comparable with best clustering result of other algorithms. Also, to obtain stable results; we have made universal definition of input parameters for all test collections, which leads us to the problem of universal term selection.

III. TEST COLLECTIONS AND QUALITY VALUE

A. Collections

In present research, we used three collections with narrow domain short texts: CICling_2002 (this collection is recognized as one of the hardest for analysis), SEPLN_CICling and EasyAbstracts; one wide domain collection: Micro4News. All collections with "gold standards" and descriptions may be found [17]. Table I contains information about gold standard and vocabulary sizes of test collections. EasyAbstracts collection contains scientific abstracts on well differentiated topics. It could be considered as medium complexity. Collection for clustering CICling_2002 and SEPLN_CICling both contain narrow domain short abstracts and their complexity for analysis is relatively high. Micro4News contains short news and its documents are longer than in other collections, also its topics are well differentiated, so the complexity is relatively low. For each collection a golden standard exists, which is a result of classification by experts and it contains 4 groups for each

TABLE I. TEST COLLECTIONS INFORMATION

Test collections	**Collection's information**		
	Cluster's topics	***Vocabulary size***	***Vocabulary size after stop words filtering***
CICLing 2002	Linguistic, Ambiguity, Lexicon, Text Processing	953	942
SEPLN CICLing	Morphological – syntactic analysis, Categorization of documents, Corpus linguistics, Machine translation	1 169	1 159
Easy Abstracts	Machine Learning, Heuristics in Optimization, Automated reasoning, Autonomous intelligent agents	2 169	1 985
Micro 4News	Sci.med, soc.religion.christian, rec.autos, comp.os.ms-windows.misc	12 785	12 286

 ISBN: 978-1-61208-175-5

collection. Collections contain 48 texts each. For our experiments test collections were additionally parsed to remove stop words.

B. Quality Values

To test quality of clustering, we use measure based on F -measure [4], we will sign it as FM :

$$FM = \sum_i \frac{G_i}{|D|} \max_j F_{ij}, \text{ where } F_{ij} = \frac{2 \cdot P_{ij} \cdot R_{ij}}{P_{ij} + R_{ij}},$$

$$P_{ij} = \frac{|G_i \cap C_j|}{|G_i|}, \; R_{ij} = \frac{|G_i \cap C_j|}{|C_j|},$$

$G = \{G_i\}_{i=\overline{1,m}}$ is an obtained set of clusters, $C = \{C_j\}_{j=\overline{1,n}}$ is set of classes, defined by experts, D - number of documents in taken collection. We use FM as quality value in this paper.

IV. ALGORITHM DESCRIPTION

A. Pearson Correlation as a Metric for Clustering

We assumed that texts in the same subject have several features that could be measured.

- There exists a group of words which always occur together in texts of one thematic group.
- Some of these words occur often in each text of a subject, some words occur rarely in each text, but all these words could be found in significant number of texts.

These assumptions lead us to the idea that if two texts have words with the same frequency characteristics, then they are semantically close to each other. Relation between texts based on the mutual word frequencies could be expressed using correlation coefficient. In our research, we present texts as N - dimension vectors, where N is the number of selected words for text representation. In our research we used Pearson correlation coefficient between two texts as a similarity function. It is calculated using formula:

$$p_{x,y} = \frac{\sum_{i=1}^{N}(x_i - M_x)(y_i - M_y)}{(N-1)\sigma_x \sigma_y},$$

where N – is a number of clustering space dimensions; x_i, y_i are values of paired variables: frequencies of a word i in document x and in document y ; M_x, M_y are values for x and y which represent average frequencies of all words in document x and y ; σ_x, σ_y - standard deviation for documents x and y .

Consider two texts test_1 and text_2 and let these texts be represented by the same set of 20 words. Consider a 2-dimension plot where horizontal and vertical axis contain frequencies of words occurrence in each of two texts. Each

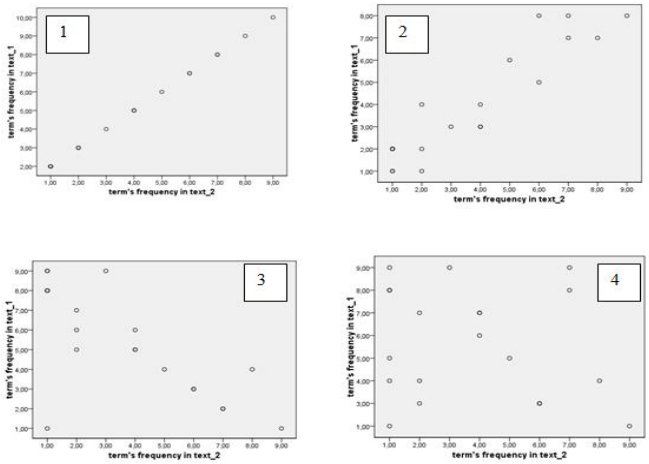


Figure 1. Pearson correlation (1: +1; 2: +0,926; 3: -0,722; 4: -0,192).

dot on such plot represents concrete word and it is placed according to frequencies in first and second texts. Four such plots are depicted in "Fig. 1". On the first plot each word of the first text occurs one more time than in the second text. In this case correlation coefficient between two texts is equals to 1. However in reality such relation is almost impossible. Second plot represents the positive relation between words: frequency characteristics of words for both texts are almost the same. But difference between frequencies of words in two texts is defined empirically and it couldn't be expressed as a function. In this case correlation coefficient is between 0 and 1. If the value of the correlation coefficient is close to 1 then more positive relation between frequencies of words in two texts is found. In the third plot, an example of negative relation is presented: if in the first text some word occurs often, then in another text this word occur rarely and vise versa. Value of correlation coefficient in this case will be from -1 to 0. On the fourth plot an example of a near zero correlation coefficient value is depicted: the relation between frequencies of words does not have significant ordered behavior.

Our research is based on the heuristic that the closer correlation coefficient between two texts is to 1, the semantically closer these texts are to each other.

Our usage of vectors as a representation for texts does not take into account the size of texts. We assume that average frequency to meet a word in text is proportional to the text size. If so, the size of text does not have much influence on correlation metric between two texts. Let we have two very similar documents d_1 and d_2 , where document d_2 is four times longer than d_1 . Let d_1 be represented by a vector (4,3,5) and document d_2 with vector (16,12,20). In this case, Pearson correlation between texts will be 1 anyway, which we interpret as semantic equivalence.

 ISBN: 978-1-61208-175-5

B. Hierarchical Clustering

We tried algorithms of hierarchic clustering such as *Between Groups Linkage* (UPGMA) [18], *Single Linkage and*, *Complete Linkage* [7]. Working scheme is the same for all of them. In the beginning each clustering object becomes a cluster. Then, on each step, two clusters with the most value of similarity between them are linked into one cluster. These steps are made until the given number of clusters is not reached. The difference between methods is in the choice of similarity function. In Single Linkage similarity between clusters is calculated as a similarity between two most similar objects in clusters. In Complete Linkage similarity between clusters is defined as a similarity between less similar objects in clusters. In Between Groups Linkage method, a mean value of similarity is calculated between each pair of objects from both clusters. Two clusters are linked if average distance between their objects is less than average distance between objects of other clusters.

Number of clusters for hierarchic clustering should be predefined and it seems like a significant disadvantage. We investigated if the result of clustering is relatively good in case the number of clusters was determined wrong. Our goal was to check which method suits the clustering task best, if the number of clusters differs from a golden standard. We calculated clustering quality with each method as an mean value of clustering results for 3-8 clusters. Experiments showed that single linkage gives bad results on all collections. We investigated if it's possible to increase clustering quality by additionally using term selection technique.

C. Terms Selection

In our research, a simple term selection method to reduce clustering space is used. Experiments showed that for Between Groups Linkage method, term selection technique, which filters words with low value of document frequency, increases the quality of clustering. Improvement of quality is observed until the number of selected terms reaches a value about 10% of initial collection vocabulary. If the number of selected words exceeds 10% limit, then clustering quality becomes worse. Our experiments also showed that filtering words with the highest values of document frequency improves clustering quality. So, we first selected about 10% of initial vocabulary terms and then from the obtained set we removed a small number of terms with the highest document frequency values. Combination of this technique with the Between Groups Linkage clustering gives best results. For Complete Linkage such term selection method could lead to further quality reduction. Based on our experiments we conclude that for narrow domain short text clustering a Between Groups Linkage method enhanced with the given term selection method is the most suitable.

D. Detection of Keywords

In our research, a simple term selection method to reduce clustering space is proposed.

After clustering was done the problem of keyword detection should be solved. We used an algorithm presented in listing in "Fig. 2" to deal with keywords. We are using three main assumptions to deal with keywords.

- If the word is semantically significant, then its occur frequency is low in most documents, but in some documents its occur frequency is high.
- If the word is significant for cluster, then it occurs in most documents of a cluster.
- If the word is significant for cluster, then the number of documents in which this word occurs, does not exceed much the size of a cluster.

```
Let D is a set of all collection's documents;
Let C is a set of clusters for annotation with keywords;
Let W is set of all words from vocabulary of collection after term selection step;
For every cluster c from C do {
    For every word w from W do {
        Let Q is a set of documents from D where occurrence number of word w
        is less then four;
        If |Q| < |c| {
            If more then |c|-α documents from c contain word w {
                Select w as a key word for cluster c;
            }
        }
    }
}
```

Figure 2. Listing of algorithm for keywords detection.

First and third rule allow filtering the commonly used words for a given collection. Second rule allows detecting words which are typical for a cluster. We defined α parameter to regulate the minimal number of documents in cluster in which a word should occur in order to be chosen as a keyword. Increasing α will reduce the number of clusters documents in which a word should be found and thus we obtain more keywords which less reflect clusters features.

V. PRESENTATION OF RESULTS

Results of our experiments are shown in Table II. For each collection we present such information: clustering quality evaluation using different number of predefined clusters (3-8); best and worst quality measure for each clustering method. This information is given for 3 cases: 1 – without initial term selection, 2 – 10% term selection, 3 – 10% term selection with filtering 3-4 terms with the highest document frequency. BGL stands for Between Groups Linkage and CL for Complete Linkage. In most cases best results are obtained for test collections with the number of clusters equal to 4, and sometimes with 3 or 5,

Using proposed algorithm we have reached good results of clustering for mentioned collections. We link this fact with the proposed combination of chosen similarity measure and term selection approach. We remove words that occur in a small number of texts and act as a noise. The description is as follows: let a word be occurring in a small number of documents. When texts are presented as N -

 ISBN: 978-1-61208-175-5

dimensional vectors, the part of vector representing a word will be like "0" in most cases and it does not affect much the correlation between texts. From the other hand there is plenty of words, which occur in a small number of texts. To leave about 10% of a collection's terms, it was enough to remove words, which occur only in 2-4 documents, most of which occur only in 1 or 2 documents. These words act as noise and they make clustering results worse. Whenever we remove 3-4 words with highest document frequencies, the actual removed words occur in half of documents, but their frequency is usually 1 (such words as: paper or based). These words act as noise and have negative influence on the result of clustering. *Between Groups Linkage* gives better results, than *Complete Linkage*, and we think it happen because test collection includes texts, which are not near the main clusters. Single linkage method tries to build one big cluster, because clusters are placed near each other and their borders are not precise.

In Table III, results of automatic topic and keywords' set detection for each cluster are presented. We also give the value of α parameter which leads to the given results. If the cluster contains small number of texts then the annotation becomes impossible. Information is given for two cases: 1) clusters from golden standard were used 2) clusters, obtained with Between Groups Linkage clustering enhanced with 10% term selection with filtering 3-4 terms with the highest document frequency were used.

Let, $w_i \in W$, $d_j \in D$, $c_k \in C$, $d_l \in D$ correspond to definitions from "Fig. 2". For the annotation process from the "Fig. 2", value of α parameter is important. This parameter is used to determine keywords: the word w_i is a keyword if it occurs at least in $|c_k| - \alpha$ documents of cluster c_k. Words, found with a small value of α, occur often in cluster and they reflect its contents. However, sometimes with the small value of α, words included in the keyword set are specific not only for concrete cluster c_k, but also for the documents of the whole collection. This problem could be solved, with introduction of limitations for w_i : w_i reflects the topic of cluster only if the number of documents, containing w_i , is less than some threshold value. For example as threshold $|c_k|$ could be taken. In this case, common words for the whole collection will not be included in resulting set (such words as: word or corpora). From the other hand, with such approach, we can loose words, which are frequent for some concrete cluster but also are in documents, outside that cluster (words like: translation or linguistic). However we found that words, which are related to topic of cluster, occur frequently in some documents, but for collection specific and common words this is not the case. We have made an assumption that for each word w_i if it relates to the topic of cluster, measures of following two points are almost equal.

- Number of documents d_j of a cluster c_k , which were not included in the set Q because w_i occurred in document d_j more than 3 times.
- Number of documents d_j, which are not included in cluster, but in the same time contain word w_i .

First and second points are balancing each other and allow finding a topic defining word despite the threshold for occurrence, even if this word occur in more than $|c_k|$ documents. Collection specific and common words do not have significant frequencies in single documents so the first point for them will not balancing with the second point. So the introduced thresholds and limitations in the annotation algorithm allow filtering most of the collection specific words without loosing the important keywords for clusters. However as the results in Table III shows us, some collection specific words still persist in the resulting keyword set, giving more challenges for future work.

VI. CONCLUSION AND FURTURE WORK

Research presented in this paper shows that for short text narrow domain collections usage of hierarchical clustering enhanced with special term selection technique could lead to good results. Comparing with other methods discussed in [2][8][12] our approach shows results which are near best and sometimes exceed them. Proposed algorithm of keywords and topic detection allows to detect words which reflect specific of each cluster. Our algorithm gives better results on well differentiated collections, but to process collections like CICling_2002 it needs improvement and this will be the subject for future work.

REFERENCES

[1] M. Alexandrov, A. Gelbukh, and P. Rosso, "An Approach to Clustering Abstracts," In Proc. 10th Int. NLDB-05 Conf., volume 3513 of Lecture Notes in Computer Science, 2005, pp. 8–13. Springer-Verlag.

[2] L. Cagnina, M. Errecalde, D. Ingaramo, and P. Rosso, "A discrete particle swarm optimizer for clustering short text corpora," BIOMA08, 2008, pp. 93–103.

[3] G. Cselle, K. Albrecht, and R. Wattenhofer "BuzzTrack: topic detection and tracking in email," IUI'07, doi:10.2011/www.arnetminer.org/viewpub.do?pid=459847

[4] A. Feng and J. Allan "Incident threading for news passages," CIKM 2009, pp. 1307-1316, doi:10.2011/www.arnetminer.org/viewpub.do?pid=1239503

[5] J, Makkonen., "Semantic Classes in Topic Detection and Tracking," 2009, Helsinki University Print, doi:10. 2011/www.doria.fi/bitstream/handle/10024/48180/semantic.pdf

[6] B. Larsen and C. Aone, "Fast and effective text mining using linear-time document clustering,". In Proc. 5th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 1999.

[7] C. D. Manning, P. Raghavan, and H. Schütze, "Introduction to Information Retrieval," Cambridge University Press, doi:10.2011/nlp.stanford.edu/IR-book/information-retrieval-book.html

[8] D. Ingaramo, M. Errecalde, and P. Rosso, "A new anttree-based algorithm for clustering Short-text corpora," Journal of CS&T, 2010.

[9] M. Errecalde., D. Ingaramo, and P. Rosso, "ITSA*: An Effective Iterative Method for Short-Text Clustering Tasks," In Proc. 23rd Int. Conf. on Industrial, Engineering & Other Applications of Applied

 ISBN: 978-1-61208-175-5

Intelligent Systems , IEA-AIE-2010, Springer-Verlag, LNAI(6096), 2011, pp. 550-559.
[10] R. Ortiz, D. Pinto, M. Tovar, and H. Jimenez-Salazar, "BUAP: An Unsupervised Approach to Automatic Keyphrase Extraction from Scientific Articles," In Proc. 5th Int. Workshop on Semantic Evaluation, ACL 2010, pp. 174–177.
[11] F. Perez-Tellez, D. Pinto, J. Cardiff, and P. Rosso "Clustering weblogs on basis of topic detection method," doi: 10.2011/users.dsic.upv.es/~prosso/resources/PerezEtAl_MCPR10.pdf
[12] D. Pinto, "Analysis of narrow-domain short texts clustering. Research report for «Diploma de Estudios Avanzados (DEA)»," Department of Information Systems and Computation, UPV, 2007, doi:10.2011/users.dsic.upv.es/~prosso/resources/PintoDEA.pdf
[13] D. Pinto, P. Rosso, and H. Jiménez, "A Self-Enriching Methodology for Clustering Narrow Domain Short Texts," Comput. J. 54(7): 1148-1165, 2011.
[14] D. Pinto, H. Jimenez-Salazar, and P. Rosso, "Clustering abstracts of scientific texts using the transition point technique," In Proc. CICLing 2006 Conf., volume 3878 of Lecture Notes in Computer Science, pp. 536–546. Springer-Verlag.
[15] S. Smith and M. Rodr´ıguez, "Clustering-based Searching and Navigation in an Online News Source," doi:10.2011/www.inf.udec.cl/~andrea/papers/ECIR06.pdf
[16] Y. Tian, W. Wang, X. Wang, J. Rao, and C. Chen, "Topic detection and organization of mobile text messages," CIKM'10, doi:10.2011/arnetminer.org/viewpub.do?pid=2898431
[17] doi:10.2011/sites.google.com/site/merrecalde/resources
[18] doi: 10.2011/www.adelaide.edu.au/acad/events/workshop/LockhartUPGMA&NJ_calculation.pdf
[19] doi:10.2011/sjsu.edu/faculty/gerstman/StatPrimer/correlation.pdf.

TABLE II. RESULTS OF CLUSTERING

Test collections	**Results of 3cases of testing**								
	1: without initial term selection			2: 10% term selection			3: 10% term selection with filtering 3-4 terms with the highest document frequency		
CICLing 2002	FM_{avg}	FM_{max}	FM_{min}	FM_{avg}	FM_{max}	FM_{min}	FM_{avg}	FM_{max}	FM_{min}
BGL	0,482	0,53	0,42	0,635	0,68	0,54	**0,645**	**0,73**	**0,59**
CL	0,508	0,54	0,48	0,503	0,56	0,45	0,5312	0,58	0,49
SEPLIN CICLing	1			2			3		
	FM_{avg}	FM_{max}	FM_{min}	FM_{avg}	FM_{max}	FM_{min}	FM_{avg}	FM_{max}	FM_{min}
BGL	0,598	0,66	0,42	0,665	0,73	0,56	**0,722**	**0,84**	**0,65**
CL	0,625	0,74	0,54	0,598	0,67	0,55	0,703	0,84	0,58
Easy Abstracts	1			2			3		
	FM_{avg}	FM_{max}	FM_{min}	FM_{avg}	FM_{max}	FM_{min}	FM_{avg}	FM_{max}	FM_{min}
BGL	0,640	0,83	0,48	0,748	0,81	0,72	**0,788**	**0,82**	**0,72**
CL	0,787	0,9	0,72	0,713	0,75	0,63	0,680	0,71	0,61
Micro4 News	1			2			3		
	FM_{avg}	FM_{max}	FM_{min}	FM_{avg}	FM_{max}	FM_{min}	FM_{avg}	FM_{max}	FM_{min}
BGL	0,832	0.89	0,75	0,868	0,96	0,79	**0,873**	0,96	0,79
CL	0,753	0,81	0,67	0,843	0,94	0,8	0,840	0,94	0,78

TABLE III. RESULTS OF OF KEYWORDS DETECTION

CICling 2002	**Clusters**			
	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Gold Standard	Document α=3	Natur α=7	Word α=6	Relat α=6
	tradit, perform, select, order, rule, document, need, larg, techniqu, automat, compar, identifi, obtain// α = 9	natur, linguist, corpu, kind, work, develop, larg, main, known, translat, obtain, provid // α=9	lexic, word, speech, part, tag, knowledg, sens, english, compar, ambigu, algorithm, disambigu, accuraci, approach, context, method // α=11	type, rule, defin, analysi, sentenc, structur, context, relat // α=9
Automati-cally clustering	Document α=5	Word α=4	No	Represent α=7
	natur, tradit, perform, select, order, rule, document, need, techniqu, experi, automat, compar, identifi, propos, algorithm, gener, discuss, evalu, represent, obtain, provid // α=11	lexic, word, corpu, inform, speech, text, on, part, differ, describ, spanish, sens, automat, compar, disambigu, accuraci, approach, dictionari, method// α = 14	No	atur, lexic, type, mean, analysi, propos, structur, context, translat, represent, relat // α=11

 ISBN: 978-1-61208-175-5

SEPLN CICLing	**Clusters**			
	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Gold Standard	Translation α=1	Syntactic α= 8	Clustering α=4	Linguistic α=6
	systems, task, automatic, order, experiments, smt, english, target, spanish, model, translation, statistical // α =8	languages, describe, grammar, parser, parsing, information, syntactic // α =11	obtained, domain, kind, short, performance, clustering, text, measures, propose, work, clusters, cluster, narrow // α =8	presents, order, resources, level, work, time, linguistic, computational, grammar, process, spanish, considered, architecture // α =9
Automati-cally clustering	Syntactic α= 11	Translation α=4	Clustering α=3	No
	grammar, parser, corpus, formalism, information, describe, syntactic // α =14	system, translation, word, machine // α =9	measure, domain, determine, kind, short, method, algorithms, clustering, propose, clusters, cluster// α =7	No

Easy Abstracts	**Clusters**			
	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Gold Standard	Objective, search α= 6	Theorem, proof, based, words, key α= 7	Agents α=6	Learning α=6
	tabu, heuristic, computational, order, optimisation, function, constraints, heuristics, objective, scheduling, multi, quality, time, search // α=8	automated, terms, theorem, system, proof, order, implemented, proving, based, words, key// α=8	communication, system, modeling, applications, semantics, flexible, independent, model, agents, information, framework, high, agent, present, work, engineering // α=9	general, classification, set, data, real, model, algorithms, function, analysis, problems, training, methods, learning, results, method, machine // α=11
Automati-cally clustering	Solution α=3	Theorem, proof α= 4	Learning α=8	Agents α=4
	heuristic, computational, algorithm, problem, solution, problems, objective, multi, quality, time, search // α=8	automated, theorem, proof, order, complete, implemented, proving, based, design, describe, words, key// α=6	general, form, class, classification, set, algorithm, support, data, real, space, model, problem, algorithms, function, analysis, problems, number, training, methods, linear, learning, results, method, machine // α =16	communication, variety, context, importance, modeling, world, semantics, flexible, independent, level, complexity, agents, models, information, notion, high, agent, effective, dynamic, formal, work, engineering // α =7

Micro 4News	**Clusters**			
	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Gold Standard	Car α=1	Windows α=1	Jesus α=1	Medical α=1
	performance, transmission, ford, road, car, sounds, suspension, tires, driving, cars, buy, mph, engine, honda, parts, bought// α= 5	software, ms, dos, running, windows, version, microsoft, user, files // α= 5	man, god, desire, spirit, acts, words, jesus, biblical, law, christians, sins, church, bible, sin, lord, christ, christian, moral // α=5	dr, study, american, news, patient, health, disease, treatment, control, national, number, related, human, year, patients, medical // α=4
Automati-cally clustering	Car α=1	Windows α=1	Jesus α=1	Medical α=1
	performance, transmission, ford, road, car, sounds, suspension, tires, driving, cars, buy, mph, engine, honda, parts, bought// α=5	software, dos, running, windows, file, version, user// α=5	man, god, desire, spirit, acts, words, jesus, biblical, law, christians, sins, church, bible, sin, lord, christ, christian, moral// α=5	dr, fax, news, patient, women, hiv, health, drug, disease, treatment, data, states, national, research, prevention, public, clinical, david, year, patients, medical, university, medicine // α=6

 ISBN: 978-1-61208-175-5

Ontology based Spreading Activation for NLP related Scenarios

Wolf Fischer
Programming Distributed Systems Lab
University of Augsburg
Augsburg, Germany
wolf.fischer@informatik.uni-augsburg.de

Bernhard Bauer
Programming Distributed Systems Lab
University of Augsburg
Augsburg, Germany
bauer@informatik.uni-augsburg.de

***Abstract*—To handle the flood of information in the modern world new technologies are needed. One problem is the handling and filtering of information itself. Semantic technologies have been named to possess the potential to at least facilitate this problem. Another difficulty is the representation of information to humans. Different algorithms and user interface concepts have been created allowing the access on a very specific type and structure of information. However the most common and natural way for humans is to use natural language. Natural Language Processing tries to analyze the syntax and semantics of language, but often delivers unsatisfying results because of the many phenomena (e.g., ambiguity) humans use while communicating. We therefore currently develop an approach, which allows us to analyze the semantic content of natural language text based on an ontology. In this paper we present a spreading activation based algorithm, which not only helps identify the correct semantic concepts for a natural language text, but also partially solves other phenomena of natural language.**

***Keywords*-semantic. spreading activation. natural language. ontology.**

I. INTRODUCTION

Ontologies have provided a comfortable way to store and perform reasoning on semantic information. Different standards have been proposed in the past of which OWL ([1], [2]) became the de facto standard. The availability of standards lead to ontologies being used even in big companies (e.g., the automotive sector). However this introduced a new problem as a new source of information is stored independently from all the other existing information. This is especially a problem for natural language documents, which contain the same or similar information as domain specific ontologies.

Currently there are no concepts or components available to bridge this gap, i.e., the gap between semantic and syntactic information (we refer to syntactic information as meaning both lexical and syntactic information). Ontologies only contain semantic information, but lack the syntactic part. On the other side documents contain a lot of information, which is stored in natural language form. Todays natural language processing components are capable of analyzing the syntactic information with a certain degree of precision. However this still leaves the question how semantic information can be gathered from the documents and how this information can be mapped to an ontology.

At the moment we are developing a prototype, which creates a model, which links a given text to an ontology (i.e. it does not extract new information from text, but try to link different types of information, i.e., natural language documents and ontologies). However there are many challenges because of the different types of ambiguities humans tend to use while writing. Many of those problems can only be solved during runtime i.e., during the analysis process. For example identifying the correct concept for a given form requires context and background knowledge. In our case this knowledge exists within an ontology. The 'easiest' case is if one word is mapped to several different concepts and one of these concepts is the correct one (e.g., 'bank' might mean the financial institute as well as a physical object, which is used for sitting). However in other cases humans tend to either use more abstract forms for what they actually mean (e.g., they refer to just 'the car' however they refer to their very own type of car). Also they could use a word, which has nothing to do with what they actually mean (e.g., in a sentence like 'I drive a red one', 'red' can be an indication towards a specific car, which is colored red). As can be seen by those simple examples there are many different cases in which it is not trivial to identify the correct meaning of a word.

We are currently developing a concept, which tries to solve this problem. A consistent meta model, which combines semantic and syntactic information at an early stage has already been presented ([3], [4]). We have developed an algorithm based on spreading activation (i.e., a marker distribution within a graph like data structure), which helps us solving exactly those problems as mentioned before. The algorithm itself is not a complete Word Sense Disambiguation (WSD) algorithm, but represents a core part of it as our WSD is based on calculating the semantic relatedness between concepts. Some more details are given in Section III.

This paper is structured as follows: Section II presents related work. Next, Section III gives a short introduction in our previous work, on which this algorithm is based on. Section IV specifies the requirements our concept has

 ISBN: 978-1-61208-175-5

to fulfill. In Section V the concept is presented, before in Section VI several examples demonstrate the working mechanism of the algorithm. The paper is concluded in Section VII.

II. Related Work

Spreading Activation is a famous approach in many different areas, e.g., in cognitive linguistics as well as WSD. The latter is closely related to our problem (as mentioned in the introduction), therefore we will especially delimit our concept from WSD approaches.

The most closely related concept to our approach seems to be that of Kleb and Abecker ([5]), which disambiguate word senses based on RDF graphs. They state homonymy and synonymy as their main problems (whereas we differenciate some more problems as stated in the introduction). Their approach does however not directly regard the problem of overgeneralization as well as words, which reference a seemingly unrelated concept at first.

Tsatsaronis et al. ([6], [7]) describe a spreading activation based approach, which uses the information from a thesauri to create a spreading activation network (SAN) for WSD. Their concept is used to disambiguate complete sentences at once. The background knowledge used is from WordNet 2. Their approach is not capable of 'guessing' better suited concepts than those, which have already been found. In ([8]) Tstsaronis et al. further evaluate the state of the art of using spreading activation for WSD. They state that concepts, which use semantic networks show the best results.

Other approaches to WSD are seen by Agirre et al. ([9]), which use a PageRank based algorithm to disambiguate word senses in the biomedical domain. Kang et al. [10] created a semi-automatic, domain independent approach to WSD (whereas we focus on specific domains). An ontology is created semi-automatically and then used for disambiguating the words of a given sentence by finding a least weighted path through the concepts within the ontology. In contrast to our approach they seem to be limited regarding the identification of the correct sense for seemingly not related words (e.g., 'red' can still refer to 'car') as they rely on WordNet only.

Spreading Activation has been used in other domains as well. Hussein et al. ([11]) used it for context adaptation. Therefore they model application domains within an ontology and after each user action an activation flow through the network filters those nodes, which are seemingly most important to the current circumstances.

III. Basics

Our approach is based on a consistent meta model combining semantic with syntactic information ([3], [4]). Our algorithm uses the semantic information available and automatically identifies the most probable concepts at hand. Based on this, syntactic structures can be mapped to specific semantic structures.

Our prototype gets as an input a natural language text, which is first being preprocessed (i.e., tokenized, POS tagged and then a syntax tree is being created). Afterwards this information is used to parse the syntax tree bottom-up and create new semantic information based on previous information. To check how existing information can be combined the algorithm takes the ontology into consideration. The focus of this paper is exactly on that step of the analysis. The algorithm is called with two or more concepts and returns a value indicating the semantic relatedness. Further it might determine concepts, which might be better suited based on the context of the original input concepts. These new concepts will then be integrated into the solution set. The best concepts with respect to a global solution are then selected as part of an evaluation in the following steps. The final result is a semantic model of the initial input text, i.e., it contains, which words of the text correspond to which concept in the ontology. Further the relations of the concepts as indicated by the text are stored in the semantic interpretation result.

The algorithm in this paper is therefore a key component within our overall analysis process and has a great influence on the outcome of the result. Its working mechanism is described in the following sections.

IV. Requirements

As mentioned previously there are several cases, in which it is difficult to identify what concepts a human might have related to. The following gives a short overview of the requirements our approach has to fulfill.

1) Analyzing text requires disambiguating the senses of each word. Therefore it is necessary to have some kind of measurement indicating if different concepts are semantically related to each other. We assume that this information helps us in solving the WSD problem. Therefore the algorithm should return a value between 0 and 1, which indicates if specific information is available within the ontology and how closely it is related. 1 should indicate that there definitely is such a relation available. 0 means that no information could be found. This is important for disambiguating synonyms and homonyms in general.
2) As humans tend to overgeneralize their expressions (e.g., instead of talking about 'E3' in Figure 1 they talk of their 'Car') our concept should be capable of identifying the most specific information possible (hyponym), i.e., if a human talks about a 'Car', but further mentions specific attributes (e.g., the color 'Red'), it is clear to his communication partner, which type of car is meant (i.e., the 'E3'). This process should be mimiced by the concept.

 ISBN: 978-1-61208-175-5

3) Humans sometimes only mention specific attributes of what they actually refer to, i.e., in contrast to the previous requirement they don't mention the 'Car' concept, but may only refer to the car by one of its attribute. An example could be 'I drive a red one'. Still the listener knows what the speaker most probably meant (a car or here again the 'E3'). The concept should try to identify and solve this problem.

The last two requirements can be summed up by saying that although some concepts might not be linked to the correct word or the semantic relation is missing between two concepts it should still be possible to identify the actually meant concepts of the user. Such a task is difficult to achieve. Usually algorithms 'only' identify the most likely concepts for a given text out of a set of directly available concepts. Since many algorithms are based on WordNet only, domain specific information might not be available, which could indicate a relation between 'Car', 'E3' and 'Red'. Statistical WSD concepts, which rely on n-grams might in some cases be capable of handling this problem. However they require that a fact has to be stated at least once in textual form to correctly disambiguate a specific context.

V. CONCEPT

The algorithm is separated into three different phases: Initialize tokens, create token flow and analyse token flow. All phases will be explained in the following sections.

A. Definitions

For our concept we need an Ontology $O := (C, R, G)$, where C is a set of concepts, R defines a set of relations between the concepts in C and G defines a set of generalizations links between the concepts in C. The algorithm is initialized using an **input** $I := (c_s, c_y, c_t, S_c)$, where c_s is the source concept, c_y is the concept of a relation, which has c_s as its source (e.g., 'Drive' would be the concept of a relation between 'Driver' and 'Vehicle') and c_t specifies the target of the relation of c_y. Finally $S_c := c_1..c_n$ is a set of further concepts, which act as additional information (context) to the spreading process. I can also consist of (c_s, c_y) or (c_s, c_t) only. S_c is always optional.

A **token container** a is defined by the tuple (c, T, act, d). a is associated with a concept $c \in O$ (this is also the ID of the token container) and a set of tokens $T := t_1..t_n$. It basically acts as a container for all the tokens, which have reached the specific concept c. It further contains an attribute act (we will refer to attributes like $a.act$ in the following), which indicates if the concept $a.c$ has been a part of the spreading activation input I (if we talk about a being part of I or another set of concepts in further references, we actually mean $a.c$, which should be contained in the corresponding concept set). d represents the depth of the tokens concept c within the ontologies generalization hierarchy. The depth value is calculated as the position of c relative to the length of the longest branch it is located in. In the following we will refer to a_s as the container of c_s, a_t as the container of c_t and a_y as the container of c_y.

A **token** t is defined by the tuple $(orig, start, pos, e, s, dir)$. $t.orig$ holds a reference to its original container (this must be a container of one of the concepts in I). Next, it contains a reference $t.start$ to the container where it originally started from (this can, but does not have to be the original container; it may also be a container whose concept is related to the concept of $t.orig$ via generalization). $t.pos$ is the container, which represents the current position of the token. $t.e$ indicates the remaining energy of the token (if the energy drops below a certain threshold this token can not spread any further). $t.s$ describes the steps the token has already traveled within the ontology. $t.dir$ defines the direction a token is traveling in. Values can be up / down (within the generalization hierarchy) or sidewards (i.e., on an association).

B. Initialize tokens

The algorithm is initialized based on each $c \in I$ with Algorithm 1 (e.g., $INIT(c_s, 1.5)$). As can be seen the initialization is based on the generalization hierarchy of the corresponding concept. All concepts of I are basically treated the same (i.e., their energy value is the same). The only exception is c_s, which receives a higher initial energy value than the remaining elements. The cause for this is that we especially want to know if there is a path from the source to the target concept. Therefore tokens from c_s receive a higher energy, which allows them to travel further.

As can be seen in algorithm 1 the initialization is done going in both generalization directions ($INITGENUP$ means that the initialization is done up the generalization hierarchy, i.e., more general elements are initialized, whereas $INITGENDOWN$ initializes more specific elements). This is done because humans tend to be ambiguous while communicating and often use more generalized terms than they actually mean (see requirement 2 in IV). Only the context of a word helps in deciding, which concept they actually refer to. Therefore, the call down the hierarchy helps to initialize all elements, which eventually are meant by a human. In contrast the call upwards initializes all those elements, which may contain the corresponding semantic information that the current concept c inherited from them. This information is necessary in order to correctly analyze the current input.

$INITGENUP$ initializes a single concept and its generalization hierarchy upwards by creating a container for every concept in the upwards generalization hierarchy and further creating the initial tokens for each of these concepts ($INITGENDOWN$ works analogously). It is important that every concept, which will be reached by a call of $INITGENUP$ in the generalization hierarchy is treated as being a part of the original input. Therefore the $a.act$

 ISBN: 978-1-61208-175-5

attribute of their containers will be set to true. The cause for this is that each of these concepts could be the carrier of the information we will later on be searching for.

Algorithm 1 Initialization

```
procedure INIT(c, ENERGY)
    INITGENUP(c, c, ENERGY, null)
    INITGENDOWN(c, c, ENERGY, null)
end procedure
```

C. Create token flow

The set of initial tokens has been created. Now the token flow itself has to be generated. The overall process is shown in algorithm 2. As can be seen the process itself is discretized in single phases. Each current token generation $T_{current}$ leads to a new token generation T_{next}, which will only be processed after every token from the current generation has been processed. This methodology is important as the $POSTPROCESS$ call initializes a back propagation mechanism. A non discretized process would yield indeterministic results.

$CREATETOKENFLOW$ gets the set of current as well as next tokens. For every single token in $T_{current}$ it does the following: First it checks if the $t.pos$, $t.dir$ and $t.e$ attributes allow a next step. If $t.dir$ is unknown, it is allowed to travel both on associations (sidewards) as well as on generalizations (up / down). A token is however not allowed to go up, if it was going down before. Also it may not go up if it was going sidewards before. The cause for these restrictions is that the tokens elseway could reach not necessary or false concepts.

Next new tokens are being generated for the next step of the current token (i.e., tokens for the relation itself as well as the target of the relation) and added to the T_{next} set. The energy of the new tokens is based on the current tokens $t.e$ attribute and is being decreased by a fixed value. However if the container of the relation has been activated (i.e., $a.act == true$), no energy will be subtracted from the energy of the new token. This process allows us to enhance the energy of paths, which are likely to be more relevant to the spreading activation input.

Next the $POSTPROCESS$ method is called. It starts the back propagation mechanism on all containers whose $a.act$ attribute is set to true and have received new tokens in the last token flow phase. Each token on such a container then gains an increase of its energy value: $t.e = t.e + (E_{MAX} - t.e) * C_e$, where E_{MAX} denotes the maximum energy a token can have and C_e is a constant factor between 0 and 1. This mechanism is recursively continued on the predecessor of this token. By activating the propagation mechanism on such containers, which are probably relevant to the input (again $a.act == true$), only such token path are strengthened, which seem to indicate the most likely results. The cause for this is that the concepts we search for are most likely closely connected (i.e., there are only few relations and therefore few steps to get from one concept to another) and also super- or subtypes of the original input concepts c_s, c_y and c_t.

Algorithm 2 Process Tokens

```
procedure PROCESSTOKENS
    while T_next.size ≠ 0 do
        T_current ← T_current ∪ T_next
        T_next ← {}
        PREPROCESS
        CREATETOKENFLOW(T_current, T_next)
        POSTPROCESS
        T_current ← {}
    end while
end procedure
```

D. Analyze token flow

The final step consists of gathering the results from the token flow process. We first start by identifying more specific elements of the actual input (see Section IV). For this we first collect all containers for every $c \in I$, which are more specific than c. Next we sort them based on the number of relevant tokens, which arrived there (i.e., tokens from concepts of I/c), their token weight (higher is better), activation times (i.e., how often the container was activated in the $POSTPROCESS$ method, more is better) and the depth of their concept (deeper is better). We then pick the best element from this list. This then is the more specific element of c_m. However in case that we find too many elements, which might be relevant to our criteria we don't pick any elements as this would contradict the idea of specifying the initial input.

All information necessary for the final result has been computed. However it might be the case that this result might not be perfect, i.e., the initial input was ambiguous (because of ambiguous statements of a human speaker, e.g., requirement three in Section IV). For such a situation we have developed a heuristic, which identifies this case and tries to identify a better solution. First there are however some restrictions to be made: Such an 'imperfect' situation can only be identified if c_s, c_y and c_t are provided in I. In other cases there would be too few information, which would lead the heuristic to imprecise decisions. Further only situations in which either c_s or c_t are wrong can be detected. For the following we will use the example from Section IV in, which case c_t is wrong (as it references 'Red' instead of 'Car').

We first collect all available associations, which are of type c_y and reference c_t. Those are stored in a list A_p. A_p is then sorted based on the weight of the associations source and target container weights (i.e., the weight of

 ISBN: 978-1-61208-175-5

the containers based on the tokens, which arrived there). Now the algorithm looks if one of the associations in A_p has a source and a target, which matches c_s or c_t: $(r.s \subseteq c_s \vee r.s \supseteq c_s) \wedge (r.t \subseteq c_t \vee r.t \supseteq c_t)$, where $r.s$ is the source concept of a relation of A_p and $r.t$ is the target concept. If this is the case the algorithm seemingly has been used on a correct input and the spreading activation is finished. If however no association of A_p matches this condition, the algorithm will be reinitialized. For this the best association of A_p (i.e., the one with the highest source and target container weights) is used because based on the current token flow this association has been marked as the best possible match. Now the spreading activation is reinitialized with a new I':

1) The 'wrong' concept (either c_s or c_t) will be replaced with the new concept ($r.s$ or $r.t$) of the best association of A_p (in our example this means that c_t 'Red' will be replaced with $r.t$ 'Car'. A more elaborate example will be given in Section VI).
2) The old element (c_s or c_t) will be added to the list of context elements, as it might provide helpful information for the next spreading activation iteration. This is done because the user might have had a reason to mention this specific concept initially therefore the concept is not thrown away, but used as a context concept).

Regarding the example from Section IV, I was $(Person, Drive, Red, \{\})$ and I' is now $(Person, Drive, Car, \{Red\})$. With I' the process is now being restarted and the same steps are applied as described before. If in this second iteration a seemingly correct result could be found the algorithm will return it. If however the conditions for starting the heuristic would match again, we stop the process. We then return the best result from both iterations. This has proved to provide good results.

Finally a value is computed, which indicates if the information we searched for exists within the ontology. There are two different cases to be distinguished:

1) The first case occurs, if the heuristic did not step in, i.e., the initial source and target elements are still the same. Then a token t from a_t is searched, which has $t.start == a.s$, i.e., it happens to have the source container as its starting position. If such a token could be found the computation of the final value depends on the average energy of the token regarding the length of the token path (excluding the generalization).
2) The second case happens if the original source or target containers have been exchanged for a new container. If this is the case, the value depends on the semantic similarity (based on a lowest common ancestor approach) between the initial c_s / c_t concepts from I and the current, 'new' c'_s / c'_t from I' concepts.

VI. Case Study

Due to the structure of our concept there are no known gold standards for our case, as existing ones like Senseval or Semeval are difficult to use for us. Senseval-2 for example provides texts, which have been annotated with WordNet 2. However WordNet is a lexical database and therefore mainly contains linguistic information, not domain relevant semantic information. Other stochastically motivated test data is not suited for our scenario at all. We can therefore not provide any elaborate statistical evaluations yet. Therefore we focus on some actual examples from our test scenario.

Our scenario currently consists of an ontology with about 100 concepts. We will show some different examples in detail in the following section. Figure 1 shows a simplified excerpt from this ontology. Its structure describes a simple car domain, which contains drivers (driving cars), different cars with different colors (E2, E3), another car E1, which has problems with its engine. Further a CEO is supposed to drive specific cars (the E2 and E3).

The first request will show the resolution of overgeneralization. 'Driver Drives E2' is supposed to detect if the concept 'Driver' is related to 'E2' using a relation of type 'Drives'. As can be seen in the picture there is a 'Drives'-relation from 'Driver' to 'Car', which is the supertype of E2. However there is also a more specific information, which could state exactly the same and in this case is even shorter: 'CEO Drives E2'. As the 'CEO' is a subconcept of 'Driver', it will be activated in the inialization phase and will itself spread tokens. As 'Drives' is also activated the token will pass with no loss of energy to 'E2'. The same is the case for the token, which will arrive at 'E2' from 'Driver'. However, this one needed more steps for its 'journey'. After the spreading activation has finished the algorithm checks every initial starting element for more concrete information. 'Driver' is the only concept, which contains a subconcept. As there are enough hints (due to backpropagation as described in Section V) that 'CEO' might be a better suited alternative to the initial request, the algorithm proposes 'CEO' as an alternative for 'Driver' to the user. As there is a direct relation available, the semantic value of the request is calculated to be 1.

A more complex request is the triple 'Driver Drives Red', i.e., a concept 'Driver' is connected to a concept 'red' using a relation of the type 'Drives' (such a request could be the case in a sentence like 'The driver drives a red one'). As can be seen in Figure 1, 'Car' is related to color and 'E3' is related to 'Red'. If the spreading activation starts the tokens will spread through the network and due to backpropagation the 'Car' concept receives a significantly higher energy than the remaining elements, as it is part of an important path between 'Driver' and 'Red' / 'Color'. As we are searching for a triple the algorithm 'sees' that there is no direct relation available between 'Driver' and 'Red'. However the 'Car'

 ISBN: 978-1-61208-175-5

element could be matching, as one relation has 'Driver' as its start concept and 'Drive' as its type. Therefore the algorithm reinitializes itself and replaces the 'Red' element with the 'Car' element ('Red' becomes a context element). In the second pass tokens from 'Driver' as well as 'CEO' will reach 'Car' as well as 'E3'. Tokens from 'E3' will reach 'Red'. Backpropagation will then again lead to an increase of energy in 'E3' and 'CEO'. As the algorithm could successfully solve the initial request it proposes 'E3' instead of 'Red' and 'CEO' instead of 'Driver'. The semantic similarity of the request however is weighted with 0.75 because we can not be absolutely sure that the user really meant 'Car' with 'Red'.

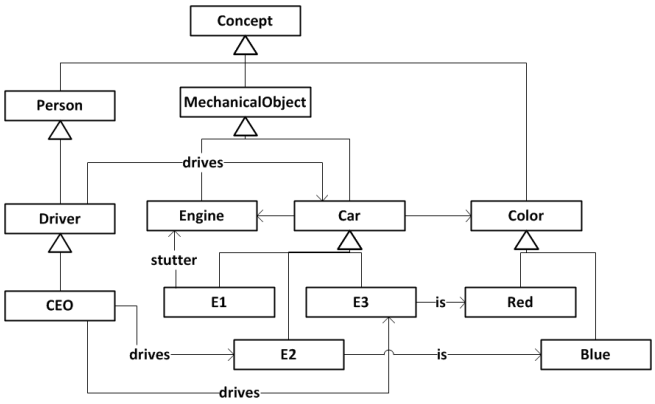


Figure 1. Example of our ontology

VII. Conclusion

The biggest problem is the knowledge acquisition problem as it is the case with every knowledge intensive system. Especially the creation of an ontology, which provides a good representation of the corresponding domain is a huge problem. We try to tackle this one by creating a corresponding set of tools and workflows, which allow an easy and semi-automatic process for this task.

In this paper we have presented a spreading activation based algorithm, which works directly on a domain ontology without creating its own SAN. It helps us in solving the WSD problem and in certain cases also proposes concepts, which are more likely to be meant instead of the initial input concepts.

The algorithm is still ongoing work and its prototypical implementation is constantly being used within our framework for creating semantic interpretations of natural language text. As such it delivers good results in our scenarios. Especially its feature of 'guessing' better suited concepts greatly helps in interpreting natural language text with all its ambiguities.

References

[1] P. Hitzler, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, "OWL 2 Web Ontology Language Primer," *Director*, no. October, pp. 1–123, 2009. [Online]. Available: http://www.w3.org/TR/2009/REC-owl2-primer-20091027/

[2] W. Ontology, "OWL 2 Web Ontology Language Document Overview," *October*, vol. 2, no. October, pp. 1–12, 2009. [Online]. Available: http://www.w3.org/TR/owl2-overview/

[3] W. Fischer and B. Bauer, "Combining Ontologies And Natural Language," *Proceedings of the Sixth Australasian Ontology Workshop*, 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/21409794

[4] W. Fischer and B. Bernhard, "Cognitive-Linguistics-based Request Answer System," in *Adaptive Multimedia Retrieval. Understanding Media and Adapting to the User*. Madrid: Springer, 2011, pp. 135–146. [Online]. Available: http://www.springerlink.com/content/l426675knx75765m/

[5] J. Kleb and A. Abecker, "Entity Reference Resolution via Spreading Activation on RDF-Graphs," *The Semantic Web Research and Applications*, vol. 6088, pp. 152–166, 2010. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-642-13486-9

[6] G. Tsatsaronis, M. Vazirgiannis, and I. Androutsopoulos, "Word Sense Disambiguation with Spreading Activation Networks Generated from Thesauri," in *IJCAI 2007*, M. M. Veloso, Ed., 2007, pp. 1725–1730. [Online]. Available: http://dblp.uni-trier.de/rec/bibtex/conf/ijcai/TsatsaronisVA07

[7] G. Tsatsaronis, I. Varlamis, and M. Vazirgiannis, "Word Sense Disambiguation with Semantic Networks," *Work*, vol. 5246, pp. 219–226, 2008. [Online]. Available: http://www.springerlink.com/content/87p101317131078t

[8] G. Tsatsaronis, I. Varlamis, and K. Nø rvåg, "An experimental study on unsupervised graph-based word sense disambiguation," *Computational Linguistics and Intelligent Text Processing*, pp. 184–198, 2010. [Online]. Available: http://www.springerlink.com/index/N577Q110122R04J6.pdf

[9] E. Agirre, A. Soroa, and M. Stevenson, "Graph-based word sense disambiguation of biomedical documents." *Bioinformatics*, vol. 26, no. 22, pp. 2889–2896, 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20934991

[10] S. Kang and J. Lee, "Ontology-based word sense disambiguation using semi-automatically constructed ontology," *MT Summit VIII Machine Translation in the Information Age Proceedings Santiago de Compostela Spain 1822 September 2001 pp181186 PDF 287KB*, 2001.

[11] T. Hussein, D. Westheide, and J. Ziegler, "Context-adaptation based on Ontologies and Spreading Activation," *Citeseer*, 2005.

 ISBN: 978-1-61208-175-5

xhRank: Ranking Entities for Semantic Web Searching

Xin He
School of Systems Engineering
University of Reading
Reading, United Kingdom
x.he@reading.ac.uk

Mark Baker
School of Systems Engineering
University of Reading
Reading, United Kingdom
mark.baker@computer.org

***Abstract*—In general, ranking entities (resources) on the Semantic Web is subject to importance, relevance, and query length. Few existing Semantic Web search systems cover all of these aspects. Moreover, many existing efforts simply reuse techniques from conventional Information Retrieval, which are not designed for Semantic Web data. This paper proposes a ranking mechanism, which includes all three categories of rankings and is tailored to Semantic Web data. Our experimental results show that this approach is effective.**

Keywords-semantic web; ranking; RDF resource; semantic search; query

I. INTRODUCTION

Semantic Web (SW) querying, in generally, involves match making, graph exploration, and ranking, which form a process pipeline. Existing approaches to ranking SW entities (resources) can be categorised into three types, based on importance, relevance, and query length respectively. Importance-based rankings [1, 2, 3, 4] rank the importance of SW resources, e.g. classes, instance resources and properties. Relevance-based rankings [1, 2, 3, 4] match keywords to SW resources. These approaches are purely based on word occurrence, and do not take into account word order and dispersion in literal phrases. Query length-based rankings [4] rank resource by following the idea that shorter queries tend to capture stronger connections between key phrases. However, we rarely see ranking schemes used in existing SW search engines that cover all of these aspects. In addition, although Information Retrieval (IR) and web algorithms, such as PageRank and TF-IDF have been adapted for application in some SW search engines, we argue that they can be further improved to be better suited for SW data.

Therefore, by analysing the limitations presented in existing research efforts and considering the specific way that SW data is stored, this paper proposes a ranking approach, namely xhRank [5]. This is a part of a SW search engine that we have developed, and is used for ranking SW resources. All relevance, importance, and query-length based rankings are included in our approach. Our experiments demonstrate that this approach is effective and that the ranking results are compliant with human perceptions.

The rest of the paper is organised as follows. We start in Section 2 with an overview of the three situations that may occur in SW searching. Section 3 introduces the xhRank approach to ranking RDF resources on the SW. This includes all relevance, importance, and query length based rankings. The evaluation of our approach is provided in Section 4. We then discuss related work in Section 5 and conclude in Section 6.

II. THE SCENARIOS IN SW SEARCHING

In SW resource searching, there are in generally three situations, in which a user input may match an instance resource that the user intends to find (Target Resource):

1) Only the target resource is matched. The user-input keywords uniquely match with the literals that directly describe the target resource. In this case, the user intends to find a resource by providing its most direct annotations.

2) The target resource and its forward neighbouring resources are matched: The user-input keywords match not only the literals that directly describe the target resource, but also the literals that describe its forward neighbours. These neighbours represent the attributes of the target resource. In this case, the user intends to find a resource by providing its most direct annotations as well as information about some attributes of the resource that is known to the user.

3) Only forward neighbouring resources of the target resource (but not the target resource itself) are matched: The user-input keywords match the literals describing the forward neighbours of the target resource, but not the literals describing the target resource itself. In this case, the user intends to find a resource by providing information about some attributes of the resource that is known to the user.

III. THE XHRANK APPROACH

In xhRank, all these situations mentioned in Section 2 are covered in the overall ranking, which is a summation of the relevance-based, importance-based, and query length-based rankings, as presented below.

A. Relevance-based Ranking

Relevance-based ranking includes Term-level, Phrase-level, and Graph-level rankings, as detailed below:

1) Term-level Ranking

In xhRank, the similarity between two terms are computed based on the Levenshtein Distance or Edit Distance

 ISBN: 978-1-61208-175-5

algorithm, which is by default supported by the Fuzzy Search functionality of Apache's Lucene [6]. According to the algorithm, the similarity between two terms (two strings) is computed depending on the minimum number of operations, e.g., an insertion, deletion, or substitution of a single character, needed to transfer one term into another.

2) Phrase-level Ranking

xhRank employs an alternative phrase ranking approach to the word occurrence-based approach used by most existing SW search systems. In addition to syntactical similarity, our approach takes into account term order and dispersion. The degree of similarity of a phrase (Key Phrase) to another phrase (Target Phrase) is determined by a phrase, called Related Key Phrase, extracted from the key phrase, in which each word corresponds to a word in the target phrase and in which the term order is compliant with the target phrase. Figure 1 illustrates a comparison example between word-occurrence and xhRank based rankings.

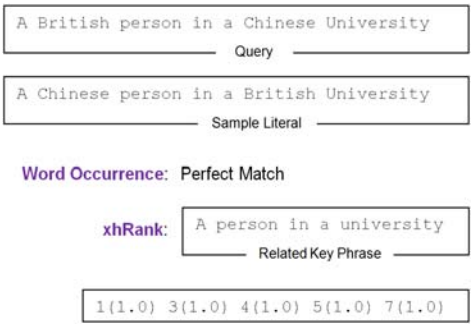


Figure 1. A comparison between word-occurrence based and xhRank based rankings

In this case, intuitively, xhRank's phrase level ranking is more reasonable than simply counting the word occurrence. Based on word occurrence, the key phrase and target phrase in Figure 1 are perfectly matched (all the seven keywords are related). However, based on human perception, we know that the query will return the wrong person in the wrong University. However, what has actually been matched is "a person in a University". In xhRank, the system finds that only five terms are related.

It should be noted that there may be more than one such related key phrase exists for a key phrase - target phase pair.

In the context of SW query, a key phrase refers to a phrase extracted from the user input, whilst a target phrase refers to the value of a literal. Instead of returning an overall score as the result, the resulting related key phrases (Phrase Similarity Result) are returned, with each word in the related key phrases represented by its position in the key phrase, in conjunction with a rating value for that word. Each word in the related key phrase is rated according to the (1) Syntactical similarity S: the similarity score between the keyword and the corresponding word in the target phrase; (2) Importance of the keywords I: specified by the user; (3) Normalisation ratio N: used to normalise the related key phrase by the length of the literal. The higher the ratio of words in the key phrase to words in the target phrase, the more valuable these words are; and (4) Discontinuous weighting D: The more times the words in the related key phrase are divided by the non-related words, the less valuable these related words are.

It should be noted that somewhat complicated algorithms are required to enable such rankings. Thus, in many cases, this technique requires more computational resources than word-occurrence based rankings. The complexity of the computation is highly dependent on the length of the target phrase. Therefore, this approach favours relatively short target phrases. It would be very costly to implement this approach on a web search, in which target phrases refer to web documents. However, in the SW paradigm, target phrases refer to literals, which are normally very short in length (in most cases less than five words). Therefore, this approach is particularly suitable for searching the SW.

3) Graph-level Ranking

This computes the degree of relevance of a graph against a user input. The graph mentioned here is the resulting graph from a graph exploration process. The node where the graph exploration initiated is called the Central Node, which is by design related to the user input, and the graph itself is called a Context Graph. Graph-level ranking is used to compute the relevance of the central node to the user input, which is subject to all resources within the context graph whose literals are related to the user input. Each of these resources is called a Related Node. For example, in Figure 2, graph A is the context graph of target node R_2. L_7, L_8 and L_9 are literals related to the user input. R_3 and R_4 are therefore related nodes.

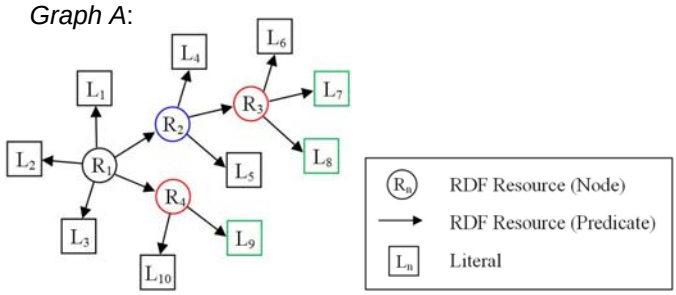


Figure 2. An example of the Context Graph of a Target Graph

The relevance of a graph to a user input is subject to the literals that are related to the user input. As related literals only describe related nodes, in other words, the relevance of a graph against a user input is subject to all related nodes within the graph. Apart from the central node, which is always a related node, these related nodes may also appear as neighbouring nodes within the context graph.

The relevance of a graph to a user input is calculated based on how well the user-input key phrases are covered by the related literal phrases within the graph. It leverages the results of phrase-level ranking, known as Phrase Similarity Result, which is a group of related key phrase lists. Each list consists of a number of elements, each of which is a keyword position and relevance score combination. Thus, against each key phrase, if there is more than one node

 ISBN: 978-1-61208-175-5

related, there may be more than one possibility of coverage as the result. By assembling these related key phrase lists for the related literal phrases, all possible coverage against a key phrase is obtained. The relevance score against a key phrase is thus computed subjects to the best coverage result. For example, Figure 3 illustrates how two related key phrase lists are assembled.

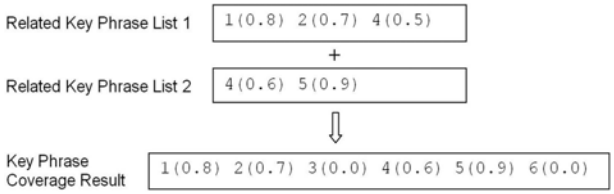


Figure 3. An example of assembling two related key phrase lists

The phrase similarity results (for all related literals) are then assembled. Figure 4 illustrates how phrase similarity results are assembled.

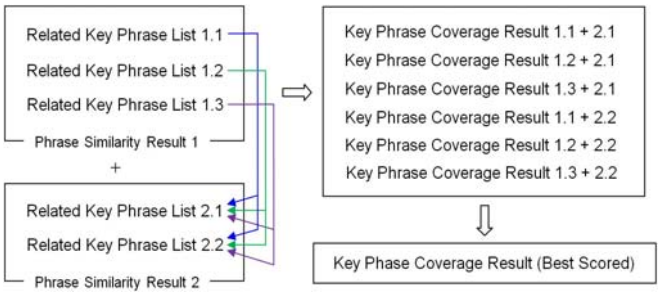


Figure 4. An example of assembling two phrase similarity results

A score against each key phrase coverage result is then calculated based on the average score of each position. The highest score among all key phrase coverage results is selected as the relevance score of the graph to that key phrase. Hence, the overall relevance score for the whole user input (including all key phrases) is calculated as the average relevance score for each key phrase.

B. Importance-based Ranking

This includes ranking the importance of SW class and instance resources (as nodes) and SW property resources (as edges) in RDF graphs.

1) Resource (Node) Ranking

The quality of resource importance rankings (based on linkage structure) depends heavily on how well the graphs and the contained RDF resources are interlinked. The ideal situation is that all resources and graphs are semantically interlinked with all related resources and graphs on a global scale, thereby forming a comprehensive graph for ranking. However, as our experiments are conducted against individual RDF datasets, resources are only linked within datasets. This will dramatically influence the ranking results. Therefore, importance ranking for SW resources is not implemented in our current experiments.

However, we still consider a variation on ReConRank [1] (the ranking approach as used in SWSE [7]) has the potential to offer an effective approach for ranking the importance of SW resources. ReConRank is a PageRank-like approach, which interconnects both resources and documents into one graph using semantic links and ranks resources based on that graph. The limitations of ReConRank are: First, the computation of the linkage-structure ranking is subject to incomplete graphs (the nodes that are related to the user input), which affects the query accuracy; Second, the ranking is performed at query time, thus affecting query speed. Therefore, by executing ReConRank-like ranking based on a complete graph (at global scale) and prior to query time, the ranking of resources' importance can be efficiently executed.

2) Property (Edge) Ranking

The importance of each property is ranked dependent on the cost of that property. This is a prerequisite of query length-based ranking, and is only applied to the properties that describe instance resources. In xhRank, the cost of a property *P* in the unit-graph of a resource *A* is determined by the popularity of *P* among all instance resources of class *C*, where *A* is an instance of *C*. Thus, each property is ranked against a class. The cost of *P* against *C* is calculated using equation (1), in which |*property*| is the number of *P* found among the instances of *C*, and *N* is the total number of instances of *C*. This is similar to the approach employed in Q2Semantic [4]. It applies to all properties including those connected with blank nodes in both directions. The lower the cost of a property, the more important the property is.

$$Cost_{p-c} = 2 - \log_2\left(\frac{|property|}{N} + 1\right) \quad (1)$$

C. Query Length-based Ranking

In xhRank, in general, query length-based rankings are used to evaluate a node (Central Node) within a graph (Context Graph) against a user input. Thus, the target node is evaluated based on the semantic distance between the target node and each of the nodes within the context graph that is related to the user input (Related Node). (See Section III A 3.)

By assuming each edge in the context graph has the same importance, the ranking score of a target node is computed as the average length of each path between the target node and a related node.

xhRank also provides an option to weight backward links lower than forward links, by altering the value of a factor called BackwardLinkRate (BLR), which is a positive number in the interval (0, 1). Hence, by considering both the importance of edges and the *BLR* factor, a target node is evaluated using equation (2), in which p_i is a path between the target node and a related node, *e* is an edge in p_i, and *n* is the quantity of such paths.

 ISBN: 978-1-61208-175-5

$$E_{t-n} = \frac{\sum_{i\in(1,n)}\left(\sum_{e\in p_i} Cost_e \times BLR\right)}{n} \quad (2)$$

D. Overall Ranking

Overall ranking extends the graph-level (relevance) ranking by complementing it with importance and query-length based rankings. The input to the ranking process is a list of explored graphs generated by the graph exploration process (a process prior to ranking). Each explored graph has a related node as its root. Thus, overall ranking is performed against each of these explored graphs (as the context graph) and against a node within the graph (as the target node). In the three situations discussed above, in situation (1) and (2), the target node is just the root node of the explored graph, which is also a related node. However, in situation (3), the target node is not a related node, but the "super-node" (backward neighbour) of all related nodes within the context graph. Thus, for each explored graph, in addition to the root node, the Top Node is also selected as a target node. A top node of an explored graph is the node, from which all related nodes can be navigated to by means of following only forward links.

In addition, there are a few points to note:

- Although explored graphs are strictly hierarchical, there can still be more than one top node in an explored graph. In this case, only the top node with the closest overall distance to the related nodes is selected.
- Top node strategy is applied only when there is more than one related node in the explore graph, which would otherwise fall into situation (1).
- Non-root related nodes in an explored graph are not selected as target nodes.

Therefore, in order to incorporate query-length based ranking into the graph-level (relevance) ranking, when performing the graph-level ranking, prior to the related key phrase lists being assembled, the relevance score for each keyword position is multiplied by the reciprocal of the cost of the path from the target node to the candidate resource described by that literal.

In order to introduce the importance-based ranking to the graph-level (relevance-based) ranking, the importance of each resource node and the cost of each property is applied to the graph-level ranking.

Hence, the overall ranking of a target node against a user input is obtained. Consequently, the overall ranking value of all target nodes are ordered, and the best K results are returned to the user.

It should be noted that graph explorations are performed based on the SW data, which includes all semantic relations that have been deduced from the corresponding ontologies prior to query time. Therefore, by interpreting the three situations (by means of following the semantic links) all semantics of the SW data are discovered.

IV. EVALUATION

We have developed a keyword-based semantic search system to demonstrate and evaluate our ranking approach. As there is currently no standard benchmark for evaluating searching against the SW, we select real world RDF datasets for our experiments. Our selection criteria are, we select RDF datasets that (1) are well known; (2) are in use; (3) are of different size; and (4) have different usage and purposes.

Based on these criteria, the datasets selected for our experiment are given below.

- myExperiment [8]
- the Lehigh University Benchmark (LUBM) (50) [9]
- DBLP (RKB Explore) [10]

(Although LUBM is a benchmark dataset, it effectively represents complicated RDF structures, and is valuable for evaluating the searching accuracy on relation based resource queries.)

We evaluate our ranking approach in terms of the system effectiveness (the accuracy of searching).

The ultimate result of the proposed semantic framework in this research will be the ranking of the available resources, indicating which is the best match, which is the next best and so on. Therefore, the objective of the effectiveness evaluation experiments is to show that the resultant matchmaking and rankings computed by the system agree reasonably well with human perception for the same situation.

A detailed study about existing effectiveness evaluation approaches has been conducted in [11], in which two basic conclusions have been drawn:

(1) There are no agreed, best practice evaluation methods that can be used to evaluate semantic matching solutions.

(2) The precision and recall metrics used in conventional IR domain cannot be directly applied to measure effectiveness of systems that return a fuzzy value for the relevance. They are only applicable to systems that return a Boolean relevance.

Therefore, we have adopted the Generalised Measures of Precision and Recall employed in [11] to evaluate their system effectiveness.

Our experiments have been carried out against the selected datasets. In line with the typical situations discussed in Section 2, we have selected six query examples, two examples for each situation, to demonstrate how the system effectively retrieves results in different scenarios. These results have been compared with human perceptions.

Participants have been selected for the human participant studies. Our selection criteria are shown below: We select human participants (1) in different age range (from 25 to 50); (2) of both male and female gender; (3) who have excellent English reading skill; (4) with different backgrounds (eastern and western); (5) with different expertise (IT including people from the Semantic Web community, Mechanical engineering, Business, Finance, Accounting, Food industry etc.)

The aim is to minimise biasing results by selecting a cross-section of participants.

 ISBN: 978-1-61208-175-5

For each query case, the user input (the keywords) is provided, followed by an explanation of what exactly the user intends to find through the query.

Top five-scored results of each query are selected for the participants to rank. These results are given in random order. The original order computed by our system is hidden to the participants. Each result is shown by a diagram illustrating the semantic relations between the matched resource and its neighbours. For the sake of simplicity, each resource (a node) is represented using the literal values (including the label values of the corresponding datatype properties) that describe the resource. Each object property (an edge connecting two resources) is represented using its label values. There is also an explanation of the diagram followed in the next page, which help the participants to capture the semantic meanings of the result.

It should be noted that each result selected for the human participant studies are scored differently from others. Where results have the same score, we randomly select one from them for the study. This is because, as our ranking system is very sensitive, query results with the same score usually have the same semantic relation structure, and have exactly the same matches to the keywords. There is little value in the participants ranking these results in order to investigate the effectiveness of our system. However, studying results with differing scores generated by our system makes it relatively straightforward to discover how accurately our system ranks the query results with different similarities to user requirements. In practise, the top-k results will be returned to the user.

The query cases are given in Table 1.

The comparisons of the system rankings and average human rankings of the query results for each query case are stipulated in Table 2.

TABLE I. QUERY CASES

Query	Scen	Keywords	User intends to find
Q1	1	matchmaking rank, semantic web, volume1	A publication. The title includes keywords "matchmaking", "rank", and "semantic web". It is published in "Volume 1" (of a Journal, for example).
Q2	1	Constraint Normal Logic Programming, Functorial Framework	A publication, which includes keyword phrases "Constraint Normal Logic Programming", and "Functorial Framework".
Q3	2	Applications of Membrane Computing, Gabriel Ciobanu 2006	A publication. The title includes key phrase "Applications of Membrane Computing". It is related to a person called "Gabriel Ciobanu". The publication year is "2006".
Q4	3	Yanchun Zhang Jinli Cao 2003	A publication between two people, called "Yanchun Zhang" and "Jinli Cao" respectively. This is published in 2003.
Q5	2	AssociateProfessor9 GraduateCourse30 Publication6	A person called "AssociateProfessor9", who is related to a graduate course, called "GraduateCourse30", and a publication entitled "Publication6".
Q6	3	Department20 University3 Course47 GraduateCourse44	A person in a department, called "Department20" at a University, called "University3". This person is related to an (undergraduate) course, called "Course47", as well as a graduate course, called "GraduateCourse44".

TABLE II. COMPARISONS OF SYSTEM AND HUMAN RANKINGS FOR QUERY RESULTS

Query Results (Ranked by System)	Average Human Ranking
1	1.36
2	3.09
3	3.00
4	3.45
5	4.09

(a) Query 1

Query Results (Ranked by System)	Average Human Ranking
1	1.27
2	3.73
3	3.18
4	3.27
5	3.55

(b) Query 2

Query Results (Ranked by System)	Average Human Ranking
1	1.27
2	3.09
3	2.45
4	3.73
5	4.45

(c) Query 3

Query Results (Ranked by System)	Average Human Ranking
1	1.64
2	1.45
3	3.27
4	4.09
5	4.54

(d) Query 4

Query Results (Ranked by System)	Average Human Ranking
1	1.18
2	2.81
3	3.54
4	3.64
5	3.81

(e) Query 5

Query Results (Ranked by System)	Average Human Ranking
1	2.00
2	2.90
3	3.36
4	2.45
5	4.27

(f) Query 6

 ISBN: 978-1-61208-175-5

The resulting generalised measures for the precision and recall against each query case are stipulated in Table 3.

TABLE III. THE PRECISION, RECALL, AND F-MEASURE FOR THE QUERY CASES

Query Case	Situation	Precision	Recall	F-measure
1	1	0.855	0.854	0.855
2	1	0.782	0.782	0.782
3	2	0.864	0.863	0.864
4	3	0.900	0.899	0.899
5	2	0.847	0.845	0.846
6	3	0.774	0.773	0.773

It should be noted that there are a number of issues that affect the participants' rankings in our experiments, as presented below.

(1) The participant's level of understanding of the Semantic Web structure. During our human participant studies, we have found that enabling ordinary users to gain an understanding Semantic Web concepts and operations presents a significant challenge. Most people are used to conventional means of gathering information, in which all retrieved data of a search result is presented in a single node (e.g., a web page). Many of the participants find it difficult to comprehend why we return a single node as a matched result, rather than the full picture shown to them. Further, in some scenarios, they may have trouble understanding why a resource is regarded as a matched resource, even if the text describing the node contains none of the keywords, whilst in other cases, resources that contain matched texts are not selected, for example the result1 of Query scenario4. This causes some confusion for users. We have tried to explain the Semantic Web as a large knowledge base. However, it seems that this explanation is still not very helpful for some participants. As the Human-Computer Interaction (HCI) implications of the Semantic Web are not the focus of this research, we have accepted that the experience for users may not be completely intuitive. Nonetheless, we have gained a deeper understanding of how significant the HCI is, and how important good interfaces are in helping ordinary people to become consumers of the Semantic Web, and in enabling them possibly to contribute to it.

(2) Familiarity with the context of the subject matter. In all six query scenarios, we require participants to rank the results according to semantic meanings rather than syntactical similarities or word occurrences. This requires the participants to understand the meanings of the textual information to a certain extent. For example, in query scenario1 and scenario2, the user intends to find a publication in the Computer Science (CS) or Artificial Intelligence (AI) domain. Some participants are not familiar with scientific phraseology, and have problems interpreting the exact meaning of the titles of publications.

(3) Human common sense does not apply. Most datasets used in the Semantic Web community are still isolated with limited inter-connections, and are mainly used for research purposes. Therefore, common sense judgements are not normally applicable to this data. For example, xhRank rankings are in part based on popularities of resources and properties. In query scenario 4, the result1 and result2 have exactly the same similarity based on relevance and query-length. The system ranks result1 over result2 because of the importance of the resources. Result1 belongs to class "Book Section Reference", whereas result2 belongs to class "Article Reference". In the DBLP dataset, there are 780,998 instances of Book Section References and 495,071 instances of Article References. Thus, an instance of class "Book Section Reference" has higher importance than an instance of class "Article Reference". However, this information is hidden to the participants, and they are unable to use common sense to interpret the rationale behind the rankings. In real-world searches, when a user searches for keyword "No.7" in amazon.com for example, it is expected that the system will rank "Chanel No.7" perfume higher than "Wilton No.7 Flower Nail", as the former product is more popular than the latter, although they have the same syntactical similarity to the keyword.

Although the above issues have encountered in our experiments, the overall results ranked by our system are still optimised. According to the human participant studies, the system is able to effectively locate the best matched result, which is most important for the users. The rest of the order of the results produced by the system is reasonably compliant with human perception.

It should be noted that this evaluation is limited by the number of people who participate in the exercise, and the amount of time they were able to devote to each study. Although the human participants are carefully selected, there will unavoidably be some bias arising from the subjective view of the participants. In addition, going through six studies takes an average of over two hours to complete. It is unavoidable that participants will tend to focus less by the time they get to the last few studies. In ideal circumstances, the system should be put on line to enable public access to the system. The evaluation should then be conducted by statistical analysis of the time each search result (the link) is clicked. This will ensure that the system effectiveness is more accurately evaluated.

V. RELATED WORK

As presented in Section 1, xhRank is employed in our SW search engine, which searches SW resources. There are numerous well known SW search systems, such as Semplore [2], Falcons [3], Q2Semantic [4], SWSE [6], Swoogle [12], Watson[13], SemSearch [14], and Sindice [15]. The majority of these systems are currently the most widely used search systems for the SW, in particular the Open Linked Data [16]. Swoogle, Sindice, and Watson are mainly used as document-oriented SW search engines, whereas Falcons, Semplore, Q2Semantic, SemSearch and, SWSE specialist in entity-oriented SW searches, which are more related to our work.

In general, ranking schemes employed in existing SW search systems can be categorised into three types, based on importance, relevance, and query length respectively. Most of these ranking schemes cover one or two categories. Importance-based ranking can be further categorised into

 ISBN: 978-1-61208-175-5

Linkage-structure based (a variation of Google's PageRank) and popularity based approach. Swoogle uses linkage-based approach to rank the importance of SW document, but not SW resources. SWSE is SW resource-oriented. However, the linkage-based approach is based on incomplete graph structure and is executed at query time, which affects the query performance and accuracy. The popularity based approaches are used by Falcons and Semplore to rank SW resource and used by Q2Semantic to rank properties. Relevance-based rankings are used by many systems, such as Falcons, SWSE, Q2Semantic, Semplore, SemSearch, and Sindice to match keywords with SW documents or resources. These approaches are purely based on word occurrence, and do not taken into account word order and dispersion within the literals. Query-length-based approaches are used by Q2Semantic to match resource. However the ranking is based on clustered (incomplete) graphs.

Compared to the ranking mechanisms implemented in existing SW search systems, xhRank covers all these three categories of ranking types; its ranking algorithm is based on complete RDF graph structures; and it supports an alternative to the conventional word occurrence approach. Experiments we have conducted show that the ranking effectiveness is very good and the ranking results are compliant with human perceptions.

VI. CONCLUSION AND FUTURE WORK

In this paper, a ranking approach, namely xhRank, is proposed, which is tailored to the nature of the SW data, in particular, the three possible situations in SW resource searching. The phrase-level (relevance-based) ranking provides a means to compute the similarity between two phrases by considering term relevance, position, and dispersion. The introduction of the importance and query length-based rankings to the graph-level (relevance-based) ranking further improves the ranking accuracy.

Our future research will begin with running our system against the Open Linked Data and Billion Triple Challenge [17], which contains the largest scale and very well interlinked SW datasets. Moreover, as explained in Section IV, an improved user interface will be developed for ordinary users to understand the query results in a more straightforward manner. Our system will be put on line to enable public access, and the evaluation will then be conducted by statistical analysis of the time each search result (the link) is clicked. These will ensure that the system effectiveness is more accurately evaluated.

ACKNOWLEDGMENT

This research is sponsored by the Research Endowment Trust Fund (RETF) and School of Systems Engineering of University of Reading.

REFERENCES

[1] A. Hogan, A. Harth, and S. Decker: ReConRank: A Scalable Ranking Method for Semantic Web Data with Context. In: Proc. 2nd SSWS, 2006.

[2] L. Zhang, Q. Liu, J. Zhang, H. Wang, Y. Pan, and Y. Yu: Semplore: An IR Approach to Scalable Hybrid Query of Semantic Web Data. In: Proc. 6th ISWC+ASWC, pp. 652-665. LNCS, vol. 4825, 2008.

[3] G. Cheng, W. Ge, and Y. Qu: Searching and Browsing Entities on the Semantic Web. In: Proc. 17th WWW, Poster Session, pp. 1101-1102, 2008.

[4] H Wang, K Zhang, Q Liu, T Tran, and Y Yu: Q2Semantic: A Lightweight Keyword Interface to Semantic Search. In: Proc. 5th ESWC. LNCS, vol. 5021, pp. 584-598, 2008.

[5] X. He and M. Baker: xhRank: Ranking Entities on the Semantic Web. In: 9th ISWC, Posters & Demo Sesstion, CEUR-WS, vol.658, pp. 41-44, 2010.

[6] Apache Lucene, URL: http://lucene.apache.org/, 20.09. 2011.

[7] A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, and S. Decker: Searching and Browsing Linked Data with SWSE: the Semantic Web Search Engine. In :Journal of Web Semantics, 2011.

[8] myExperiment, http://www.myexperiment.org/, 20.09. 2011.

[9] Y. Guo, Z. Pan, and J. Heflin: LUBM: A Benchmark for OWL Knowledge Base Systems. In: J. of Web Semantics, vol. 3, no. 2-3, pp. 158-182, 2005.

[10] DBLP (RKB Explore), http://dblpp.rkbexplorer.com/, 20.09. 2011.

[11] A. Bandara: Semantic Description and Matching of Services for Pervasive Environments. PhD Thesis, University of Southampton, 2008.

[12] L. Ding, T. Finin, A. Joshi, Y. Peng, R. Cost, J. Sachs, R. Pan, P. Reddivari, and V. Doshi: Swoogle: A Search and Metadata Engine for the Semantic Web. Proc. 13th ACM Conf. on Information and Knowledge Management, pp. 652-659, 2004.

[13] M. d'Aquin, M. Sabou, M. Dzbor, C. Baldassarre, L. Gridinoc, S. Angeletou, and E. Motta: WATSON: A Gateway for the Semantic Web. Proc. 4th ESWC, Poster Session, 2007.

[14] Y. Lei, V. Uren, and E. Motta, "SemSearch: A Search Engine for the Semantic Web", In: Proc. EKAW, pp. 238-245, 2006.

[15] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello: Sindice.com: A Document-Oriented Lookup Index for Open Linked Data. In: Journal of Metadata, Semantics and Ontologies, vol.3, no.1, pp. 37-52, 2008.

[16] Open Linked Data, URL: http://linkeddata.org/, 20.09. 2011.

[17] Billion Triple Challenge 2011 Dataset, http://challenge.semanticweb.org/, 20.09. 2011.

 ISBN: 978-1-61208-175-5

Query Expansion for Peculiar Images by Web-extracted Hyponyms

Shun Hattori
School of Computer Science
Tokyo University of Technology
1404-1 Katakura-machi, Hachioji, Tokyo 192-0982, Japan
Email: hattori@cs.teu.ac.jp

***Abstract*—Most researches on Image Retrieval have aimed at clearing away noisy images and allowing users to retrieve only acceptable images for a target object specified by its object-name. We have become able to get enough acceptable images of a target object just by submitting its object-name to a conventional keyword-based Web image search engine. However, because the search results rarely include its uncommon images, we can often get only its common images and cannot easily get exhaustive knowledge about its appearance. As next steps of Image Retrieval, it is very important to discriminate between "Typical Images" and "Peculiar Images" in the acceptable images, and moreover, to collect many different kinds of peculiar images exhaustively. This paper proposes a novel method to search the Web for peculiar images by expanding or modifying a target object-name with its hyponyms extracted from the Web by text mining techniques, and validates its precision by comparing with Google Image Search.**

***Keywords*-image retrieval; query expansion; peculiar images; hyponymy; concept hierarchy**

I. INTRODUCTION

In recent years, various demands have arisen in searching the Web for images as well as documents (text) to utilize them more effectively. When a name of a target object is given by a user, the main goal of conventional keyword-based Web image search engines such as Google Image Search [1] and most researches on Image Retrieval (IR) is to allow the user to clear away noisy images and retrieve only the acceptable images for the target object-name, which just include the target object in their content, as precisely as possible. However, the acceptable images for the quite same object-name are of great variety. Therefore, we sometimes want to retrieve not only vague acceptable images of a target object but also its niche images, which meet some kind of additional requirements. One example of more niche image searches allows the user to get special images of the target object with the impression [2–4].

Another example of more niche demands, when only a name of a target object is given, is to search the Web for its "Typical Images" [5] which allow us to adequately figure out its typical appearance features and easily associate themselves with the correct object-name, and its "Peculiar Images" [6–8] which include the target object with not common (or typical) but eccentric (or surprising) appearance features. For instance, most of us would uppermost associate "sunflower" with "yellow one", "cauliflower" with "white one", and "sapphire" with "blue one", while there also exist "red sunflower" or "black one" etc., "purple cauliflower" or "orange one" etc., and "yellow sapphire" or "pink one" etc. When we exhaustively want to know all the appearances of a target object, information about its peculiar appearance features is very important as well as its common ones.

Conventional Web image search engines are mostly Text-Based Image Retrievals by using the filename, alternative text, and surrounding text of each Web image. When such a text-based condition as a name of a target object is given by a user, they give the user the retrieval images which meet the text-based condition. It has become not difficult for us to get typical images as well as acceptable images of a target object just by submitting its object-name to a conventional keyword-based Web image search engine and browsing the top tens of the retrieval results, while peculiar images rarely appear in the top tens of the retrieval results. As next steps of IR in the Web, it is very important to discriminate between "Typical Images" and "Peculiar Images" in the acceptable images, and moreover, to collect many different kinds of peculiar images as exhaustively as possible.

My previous works [6], [7] have proposed a basic method to search the Web for peculiar images of a target object whose name is given as a user's original query, by expanding the original query with its peculiar appearance descriptions (e.g., color-names) extracted from the Web by text mining techniques [9], [10] and/or its peculiar image features (e.g., color-features) converted from the Web-extracted peculiar color-names. And to make the basic method more robust, my previous work [8] has proposed a refined method equipped with cross-language (translation between Japanese and English) functions like [11], [12]. As another solution, this paper proposes a novel method to search the Web for peculiar images by expanding or modifying a target object-name (of an original query) with its hyponyms extracted from the Web by using not hand-made concept hierarchies such as WordNet [13] but enormous Web documents and text mining techniques.

The remainder of this paper is organized as follows. Section II explains my proposed method for Peculiar Image Search. Section III shows several experimental results to validate its precision. Last, Section IV concludes this paper.

 ISBN: 978-1-61208-175-5

II. METHOD

This section explains my proposed method to precisely search the Web for "Peculiar Images" of a target object whose name is given as a user's original query, by expanding the original query with its hyponyms extracted from the Web by text mining techniques.

Figure 1 gives an overview of my Peculiar Image Search (PIS) based on Web-extracted hyponym relations, while Figure 2 gives an overview of my previous Peculiar Image Search based on Web-extracted color-names [6–8].

Step 1. Hyponym Extraction

When a name of a target object as an original query is given by a user, its hyponyms are automatically extracted from exploding Web documents about the target object by text mining techniques [14], [15]. Of course, they could be extracted from hand-made concept hierarchies such as WordNet [13]. The latter is precision-oriented, while the former is rather recall-oriented. Therefore, this paper adopts the former as a solution of the 2nd next step of Image Retrieval to collect many different kinds of peculiar images as exhaustively as possible.

The PIS system collects candidates for hyponyms of a target object o by using two kinds of lexico-syntactic patterns "a * o" and "the * o" where "*" is wild-card. Next, it filters out "* o" whose frequency of Web documents searched by submitting ["* o"] as a query to Google Web Search [16] is less than 10, and uses only the top 100 (at most) candidates ordered by their document frequency.

Step 2. Query Expansion by Hyponyms

Here, we have two kinds of clues to search the Web for peculiar images: not only a target object-name o (text-based condition) as an original query given by a user, but also its hyponyms h (text-based condition) automatically extracted from not hand-made concept hierarchies such as WordNet but the whole Web in Step 1.

The original query (q0 = text: ["o"] & content: null) can be modified or expanded by its hyponym h as follows:

q1 = text: ["h"] & content: null,
q2 = text: ["o" AND "h"] & content: null.

This paper adopts more conditioned latter to precisely search the Web for its acceptable images and "Peculiar Images".

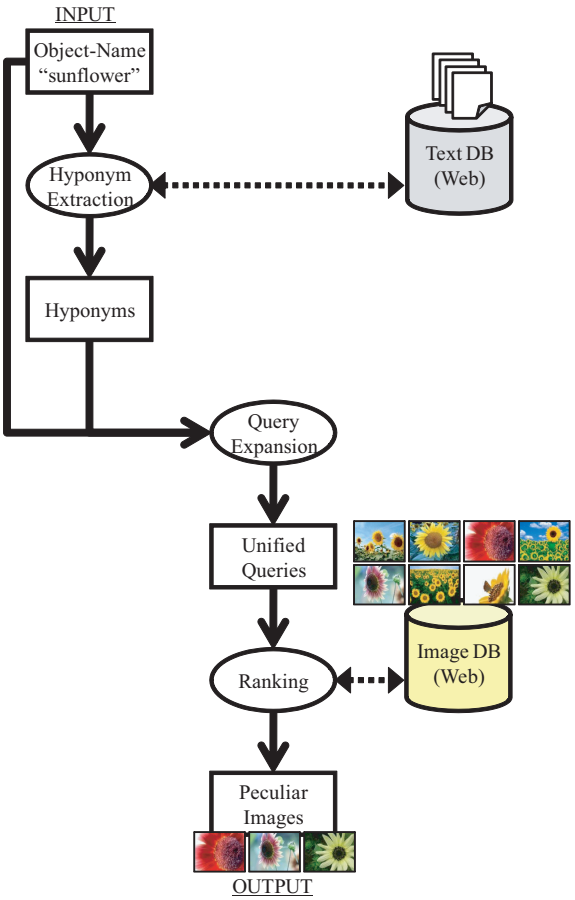


Figure 1. Peculiar Image Search based on Web-extracted Hyponyms.

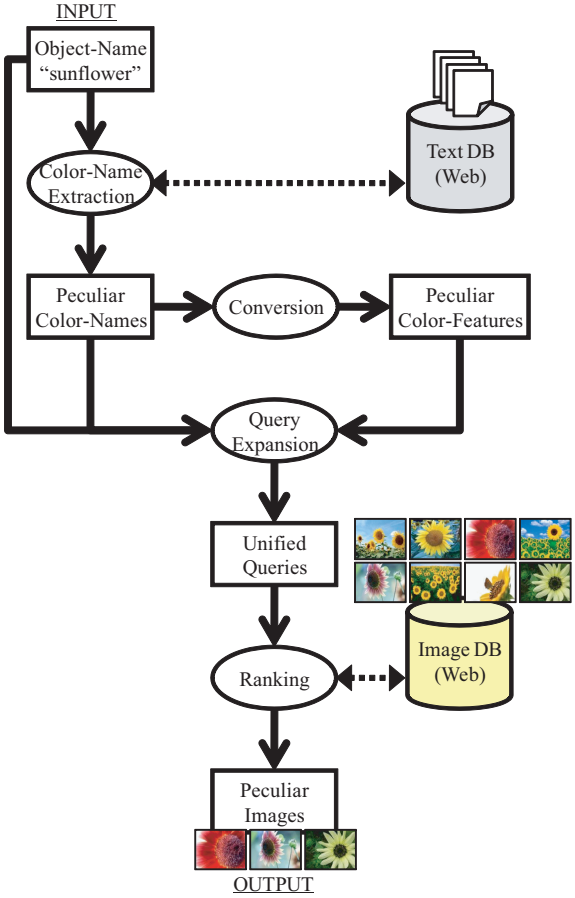


Figure 2. Peculiar Image Search based on Web-extracted Color-Names.

 ISBN: 978-1-61208-175-5

Step 3. Image Ranking by Expanded Queries

This paper defines two kinds of weights of Peculiar Image Search based on the expanded query (q2 = text: ["h" AND "o"] & content: null) in Step 2.

The first weight $\text{pis}_1(i,o)$ is assigned to a Web image i for a target object-name o and is defined as

$$\text{pis}_1(i,o) := \max_{\forall h \in H(o)} \left\{ \frac{\text{hyponym}(h,o)}{\text{rank}(i,o,h)^2} \right\}$$

where $H(o)$ stands for a set of hyponyms of a target object-name o extracted from the whole Web or the hand-made WordNet in Step 1, a Web image i is retrieved by submitting the text-based query ["o" AND "h"] (e.g., ["sunflower" AND "evening sun"]) to Google Image Search [1], and $\text{rank}(i,o,h)$ stands for the rank (positive integer) of a Web image i in the retrieval results from the Google's image database. And $\text{hyponym}(h,o) \in [0,1]$ stands for the weight of a candidate h for hyponyms of a target object-name o. In this paper, for any hyponym candidates h of a target object-name o extracted from hand-made (so certainly precise) concept hierarchies such as WordNet, $\text{hyponym}(h,o)$ is set to 1. Meanwhile, for Web-extracted hyponym candidates h of a target object-name o, $\text{hyponym}(h,o)$ is calculated as,

$$\text{hyponym}(h,o) := \text{df}(\text{["}h\text{"]}) \;/\; \max_{\forall h \in H(o)} \{\text{df}(\text{["}h\text{"]})\}$$

where $\text{df}(\text{[}q\text{]})$ stands for the frequency of Web documents searched by submitting a query q to Google Web Search.

The second weight $\text{pis}_2(i,o)$ is assigned to a Web image i for a target object-name o and is defined as

$$\text{pis}_2(i,o) := \max_{\forall h \in H(o)} \left\{ \frac{\text{ph}(h,o)}{\text{rank}(i,o,h)} \right\}$$

where $\text{ph}(h,o) \in [0,1]$ stands for the weight of a candidate h for Peculiar(-colored) Hyponyms of an object-name o,

$$\text{ph}(h,o) := \frac{(\text{ph}^*(h,o) - \min(o))^2}{(\max(o) - \min(o))^2}$$

$$\text{ph}^*(h,o) := \frac{|I_k(o)| \cdot |I_k(o,h)| \cdot \sqrt{\text{hyponym}(h,o)}}{\sum_{i \in I_k(o)} \sum_{j \in I_k(o,h)} \text{sim}(i,j)}$$

$$\max(o) := \max_{\forall h}\{\text{ph}^*(h,o)\}, \quad \min(o) := \min_{\forall h}\{\text{ph}^*(h,o)\}$$

where $I_k(o)$ and $I_k(o,h)$ stand for a set of the top k (at most 100) Web images retrieved by submitting the text-based query ["o"] (e.g., ["sunflower"]) and ["o" AND "h"] (e.g., ["sunflower" AND "evening sun"]) to Google Image Search, respectively. And $\text{sim}(i,j)$ stands for the similarity between Web images i and j in the HSV color space [17] as a cosine similarity,

$$\text{sim}(i,j) := \frac{\sum_{\forall c} \text{prop}(c,i) \cdot \text{prop}(c,j)}{\sqrt{\sum_{\forall c} \text{prop}(c,i)^2} \sqrt{\sum_{\forall c} \text{prop}(c,j)^2}}$$

where c stands for any color-feature in the HSV color space where 12 divides for Hue, 5 divides for Saturation, and 1 divide for Value (Brightness), and $\text{prop}(c,i)$ stands for the proportion of a color-feature c in a Web image i.

III. EXPERIMENT

This section shows several experimental results for the following six kinds of target object-names to validate my proposed method to search the Web for their peculiar images more precisely than conventional Web image search engines such as Google Image Search. Table I shows the numbers of WordNet's and Web-extracted hyponyms for each object.

Table I
NUMBER OF WORDNET'S AND WEB-EXTRACTED HYPONYMS.

Object-Name	WordNet's	Web-extracted	
sunflower	19	100	(of 531)
cauliflower	0	100	(of 368)
praying mantis	0	100	(of 253)
tokyo tower	0	92	(of 157)
nagoya castle	0	23	(of 57)
wii	0	100	(of 297)

Figure 3 shows the top k average precision of my proposed Peculiar Image Searches (PIS) based on Web-extracted hyponyms or hand-made concept hierarchies such as WordNet, and Google Image Search for the above-mentioned six target object-names. It shows that my PIS method by using the second (more refined) ranking $\text{pis}_2(i,o)$ is superior to my PIS method by using the first (simpler) ranking $\text{pis}_1(i,o)$ as well as Google Image Search, and that my PIS method by using Web-extracted hyponym relations is superior to my PIS method by using WordNet's ones.

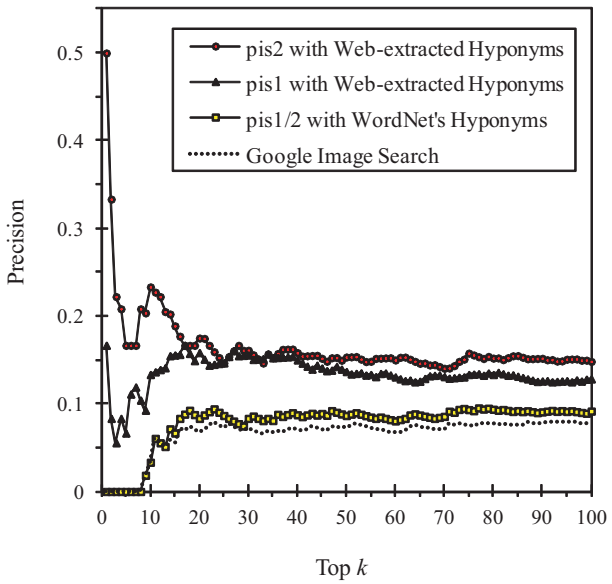


Figure 3. Top k Average Precision of Google Image Search (query: q0) vs. Peculiar Image Searches (query: q2, ranking: pis_1 or pis_2).

Table II
TOP 20 PECULIAR(-COLORED) HYPONYMS OF "SUNFLOWER".

	hyponym(h,o)		ph(h,o)	
1	good sunflower	1.000	**pink sunflower**	1.000
2	tall sunflower	1.000	raw sunflower	0.789
3	ground sunflower	0.984	shelled sunflower	0.770
4	same sunflower	0.968	brunning sunflower	0.758
5	few sunflower	0.964	roasted sunflower	0.669
6	small sunflower	0.929	complex sunflower	0.645
7	first sunflower	0.915	hotel sunflower	0.533
8	giant sunflower	0.913	**purple sunflower**	0.511
9	raw sunflower	0.910	**green sunflower**	0.493
10	growing sunflower	0.900	**black sunflower**	0.470
11	new sunflower	0.900	black oil sunflower	0.386
12	huge sunflower	0.898	gray sunflower	0.370
13	black oil sunflower	0.890	modern sunflower	0.357
14	complex sunflower	0.890	**metal sunflower**	0.335
15	brunning sunflower	0.878	emmanuelle sunflower	0.332
16	large sunflower	0.876	dried sunflower	0.331
17	toasted sunflower	0.875	given sunflower	0.289
18	tiny sunflower	0.868	**blue sunflower**	0.282
19	normal sunflower	0.856	**red sunflower**	0.277
20	u.s. sunflower	0.855	kids' sunflower	0.223

Table III
TOP 20 PECULIAR(-COLORED) HYPONYMS OF "CAULIFLOWER".

	hyponym(h,o)		ph(h,o)	
1	spicy cauliflower	1000	**purple cauliflower**	1.000
2	grated cauliflower	1.000	**pink cauliflower**	0.455
3	remaining cauliflower	1.000	fried cauliflower	0.268
4	**purple cauliflower**	0.984	spicy cauliflower	0.255
5	blanched cauliflower	0.975	**yellow cauliflower**	0.234
6	creamy cauliflower	0.975	few cauliflower	0.230
7	leftover cauliflower	0.965	huge cauliflower	0.230
8	fried cauliflower	0.948	grated cauliflower	0.191
9	raw cauliflower	0.948	regular cauliflower	0.186
10	boiled cauliflower	0.944	curried cauliflower	0.179
11	huge cauliflower	0.940	tiny cauliflower	0.168
12	**yellow cauliflower**	0.934	**golden cauliflower**	0.166
13	organic cauliflower	0.932	crispy cauliflower	0.148
14	crunchy cauliflower	0.928	little cauliflower	0.140
15	or cauliflower	0.905	tandoori cauliflower	0.139
16	baby cauliflower	0.904	**cheddar cauliflower**	0.129
17	tiny cauliflower	0.898	leftover cauliflower	0.123
18	**golden cauliflower**	0.884	yummy cauliflower	0.120
19	garlic cauliflower	0.877	larger cauliflower	0.116
20	drained cauliflower	0.874	braised cauliflower	0.115

Tables II and III show the top 20 peculiar hyponyms with peculiar color-features of a target object-name, "sunflower" and "cauliflower", respectively. They show that ph(h,o) used by the second (more refined) ranking $\text{pis}_2(i,o)$ is superior to hyponym(h,o) used by the first (simpler) ranking $\text{pis}_2(i,o)$ as a weighting function of peculiar hyponyms h for each target object-name o. Figure 4 shows the top k average precision of hyponym extraction from the Web. ph(h,o) gives 42.5% (not much different) precision at $k = 20$ for hyponym extraction, while hyponym(h,o) gives 42.5% precision. And Figure 5 shows the top k average precision of peculiar hyponym extraction from the Web. ph(h,o) gives 16.7% (superior) precision at $k = 20$ for peculiar hyponym extraction, while hyponym(h,o) gives 10.0% precision.

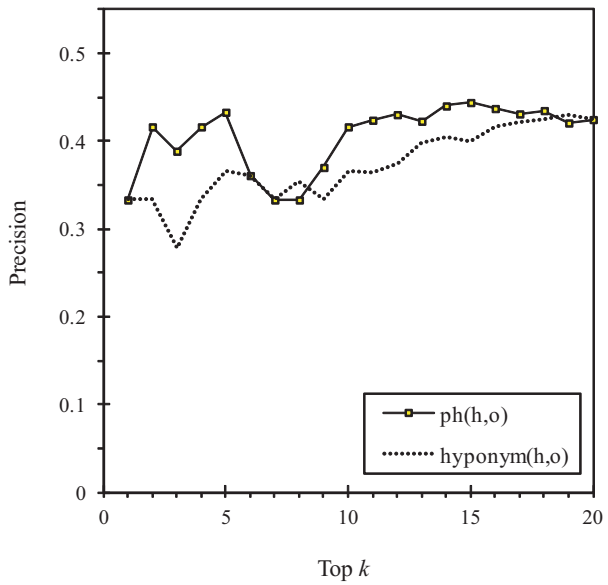


Figure 4. Top k Average Precision of Hyponym Extraction from the Web.

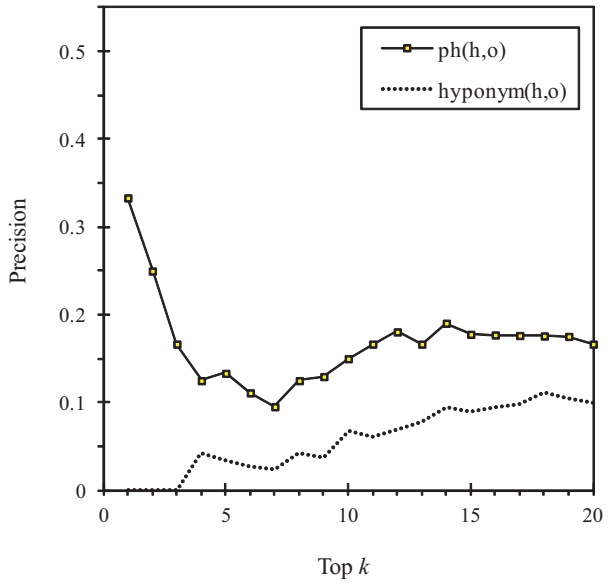


Figure 5. Top k Average Precision of Peculiar(-Colored) Hyponym Extraction from the Web.

Figures 6 to 11 show the top 20 search results for each target object-name, "sunflower" or "cauliflower", to compare between Google Image Search [1] as a conventional keyword-based Web image search engine, and my proposed Peculiar Image Search by using the first (simpler) ranking function $\text{pis}_1(i,o)$ or the second (more refined) ranking function $\text{pis}_2(i,o)$ based on Web-extracted hyponym relations. They show that my proposed Peculiar Image Searches are superior to Google Image Search to search the Web for peculiar images of a target object-name.

 ISBN: 978-1-61208-175-5

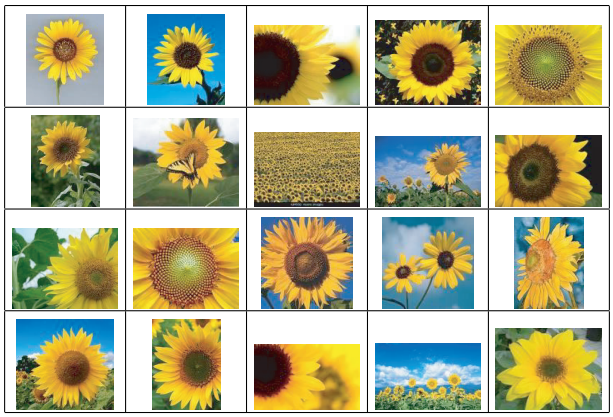


Figure 6. Top 20 results of Google Image Search (query: q0, ranking: Google, object-name: “sunflower”).

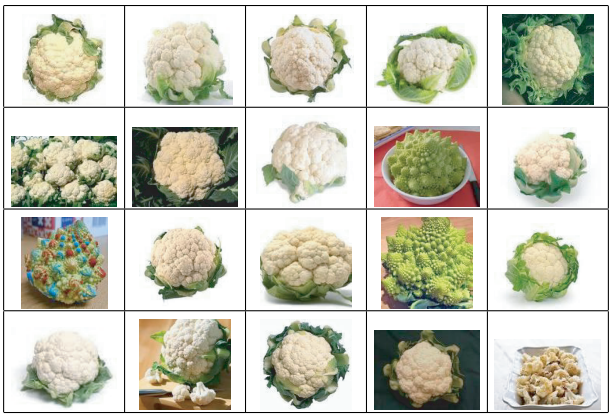


Figure 9. Top 20 results of Google Image Search (query: q0, ranking: Google, object-name: “cauliflower”).

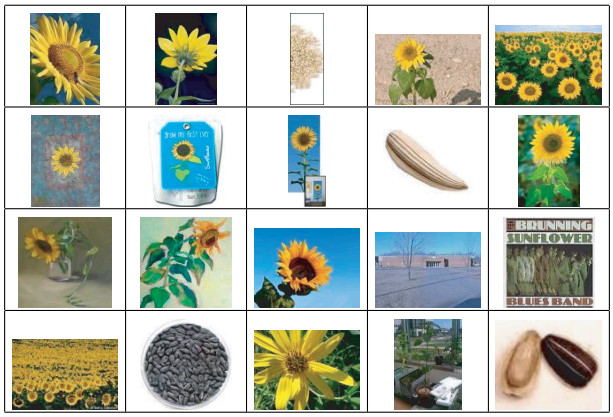


Figure 7. Top 20 results of Peculiar Image Search (query: q2, ranking: $\mathrm{pis}_1(i,o)$, object-name: “sunflower”).

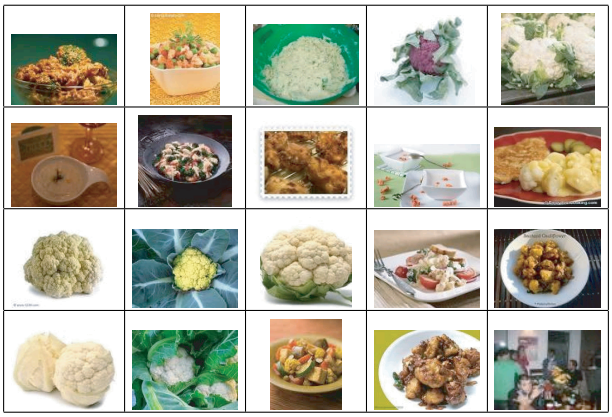


Figure 10. Top 20 results of Peculiar Image Search (query: q2, ranking: $\mathrm{pis}_1(i,o)$, object-name: “cauliflower”).

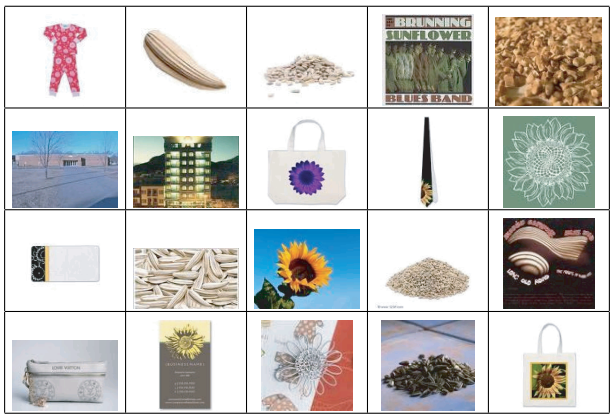


Figure 8. Top 20 results of Peculiar Image Search (query: q2, ranking: $\mathrm{pis}_2(i,o)$, object-name: “sunflower”).

Figure 11. Top 20 results of Peculiar Image Search (query: q2, ranking: $\mathrm{pis}_2(i,o)$, object-name: “cauliflower”).

 ISBN: 978-1-61208-175-5

IV. Conclusion and Future Work

As next steps of Image Retrieval (IR), it is very important to discriminate between "Typical Images" and "Peculiar Images" in the acceptable images, and moreover, to collect many different kinds of peculiar images exhaustively. In other words, "Exhaustiveness" is one of the most important requirements in the next IR. As a solution, my previous works proposed a basic method to precisely search the Web for peculiar images of a target object by its peculiar appearance descriptions (e.g., color-names) extracted from the Web and/or its peculiar image features (e.g., color-features) converted from them. And to make the basic method more robust, my previous work proposed a refined method equipped with cross-language (translation between Japanese and English) functions.

As another solution, this paper has proposed a novel method to search the Web for peculiar images by expanding or modifying a target object-name (of an original query) with its hyponyms extracted from the Web by using not hand-made concept hierarchies such as WordNet but enormous Web documents and text mining techniques. And several experimental results have validated the retrieval precision of my proposed method by comparing with such a conventional keyword-based Web image search engine as Google Image Search. They also show that my second (more refined) ranking $\mathrm{pis}_2(i, o)$ is superior to my first (simpler) ranking $\mathrm{pis}_1(i, o)$, and that using Web-extracted hyponym relations is superior to using hand-made WordNet's ones.

In the near future, as clues of query expansion for Peculiar Images of a target object-name, I try to utilize both its Web-extracted hyponym relations and hand-made concept hierarchies, and also both its hyponyms and appearance descriptions (e.g., color-names). In addition, I try to utilize the other appearance descriptions (e.g., shape and texture) besides color-names and the other image features besides color-features in my various Peculiar Image Searches.

Acknowledgment

This work was supported in part by JSPS Grant-in-Aid for Young Scientists (B) "A research on Web Sensors to extract spatio-temporal data from the Web" (23700129, Project Leader: Shun Hattori, 2011-2012).

References

[1] Google Image Search, http://images.google.com/ (2011).

[2] Robert Inder, Nadia Bianchi-Berthouze, and Toshikazu Kato: "K-DIME: A Software Framework for Kansei Filtering of Internet Material," Proceedings of the 1999 IEEE International Conference on Systems, Man and Cybernetics (SMC'99), Vol.6, pp.241–246 (1999).

[3] Takio Kurita, Toshikazu Kato, Ikumi Fukuda, and Ayumi Sakakura: "Sense Retrieval on a Image Database of Full Color Paintings," Transactions of Information Processing Society of Japan (IPSJ), Vol.33, No.11, pp.1373–1383 (1992).

[4] Haruo Kimoto: "An Image Retrieval System Using Impressional Words and the Evaluation of the System," Transactions of Information Processing Society of Japan (IPSJ), Vol.40, No.3, pp.886–898 (1999).

[5] Shun Hattori and Katsumi Tanaka: "Search the Web for Typical Images based on Extracting Color-names from the Web and Converting them to Color-Features," Letters of Database Society of Japan, Vol.6, No.4, pp.9–12 (2008).

[6] Shun Hattori and Katsumi Tanaka: "Search the Web for Peculiar Images by Converting Web-extracted Peculiar Color-Names into Color-Features," IPSJ Transactions on Databases, Vol.3, No.1 (TOD45), pp.49–63 (2010).

[7] Shun Hattori: "Peculiar Image Search by Web-extracted Appearance Descriptions," Proceedings of the 2nd International Conference on Soft Computing and Pattern Recognition (SoCPaR'10), pp.127–132 (2010).

[8] Shun Hattori: "Cross-Language Peculiar Image Search Using Translation between Japanese and English," Proceedings of the 2011 First IRAST International Conference on Data Engineering and Internet Technology (DEIT'11), pp.418–424 (2011).

[9] Shun Hattori, Taro Tezuka, and Katsumi Tanaka: "Extracting Visual Descriptions of Geographic Features from the Web as the Linguistic Alternatives to Their Images in Digital Documents," IPSJ Transactions on Databases, Vol.48, No.SIG11 (TOD34), pp.69–82 (2007).

[10] Shun Hattori, Taro Tezuka, and Katsumi Tanaka: "Mining the Web for Appearance Description," Proc. of the 18th International Conference on Database and Expert Systems Applications (DEXA'07), LNCS Vol.4653, pp.790–800 (2007).

[11] Oren Etzioni, Kobi Reiter, Stephen Soderland, and Marcus Sammer: "Lexical Translation with Application to Image Search on the Web," Proc. of Machine Translation Summit XI (2007).

[12] Jin Hou, Dengsheng Zhang, Zeng Chen, Lixing Jiang, Huazhong Zhang, and Xue Qin: "Web Image Search by Automatic Image Annotation and Translation," Proceedings of the 17th International Conference on Systems, Signals and Image Processing (IWSSIP'10), pp.105–108 (2010).

[13] WordNet, http://wordnetweb.princeton.edu/ (2011).

[14] Marti A. Hearst: "Automatic Acquisition of Hyponyms from Large Text Corpora," Proceedings of the 14th International Conference on Computational Linguistics (COLING'92), pp.539–545 (1992).

[15] Shun Hattori and Katsumi Tanaka: "Extracting Concept Hierarchy Knowledge from the Web based on Property Inheritance and Aggregation," Proceedings of the 7th IEEE/WIC/ACM International Conference on Web Intelligence (WI'08), pp.432–437 (2008).

[16] Google Web Search, http://www.google.com/ (2011).

[17] John R. Smith and Shih-Fu Chang: "VisualSEEk: A Fully Automated Content-Based Image Query System," Proceedings of the 4th ACM International Conference on Multimedia (ACM Multimedia'96), pp.87–98 (1996).

 ISBN: 978-1-61208-175-5

Towards an Ontology for Enterprise Knowledge Management

Eckhard Ammann
School of Informatics
Reutlingen University, Reutlingen, Germany
Eckhard.Ammann@Reutlingen-University.de

Ismael Navas-Delgado, José F. Aldana-Montes
E.T.S.I. Informática
University of Málaga, 29071 Málaga, Spain
{ismael, jfam}@lcc.uma.es

***Abstract*—Enterprise knowledge management is about approaches, methods, and techniques, which will support the management of the resource "knowledge" in an enterprise for the purpose of support and advancement of businesses. An important part of it is knowledge development of individual and organizational knowledge. This paper provides an overall conception of enterprise knowledge management in the form of a layered set of ontologies, which are enriched by appropriate rule systems. This set consists of general (i.e. enterprise-independent) and of enterprise-specific ontologies. General ontologies in this set include ontologies for knowledge and knowledge development and for human interaction. Enterprise-specific ontologies formalize specific domains in the enterprise as well as managerial principles and finally a whole enterprise.**

Keywords*—*Knowledge management ontology, knowledge development, organizational learning, human interaction, managerial and enterprise ontology.

I. INTRODUCTION

Enterprise knowledge management is about approaches, methods, and techniques, which will support the management of the resource knowledge in an enterprise for the purpose of support and advancement of businesses. An important part of it is knowledge development of individual, group, and organizational knowledge. Several approaches for knowledge management exist, one of them is the process-oriented approach see [1], [12], and [14]. One specific approach for enterprise knowledge development is EKD (Enterprise Knowledge Development), which aims at articulating, modeling and reasoning about knowledge, which supports the process of analyzing, planning, designing, and changing your business; see [7] and [9] for a description of EKD. EKD does not provide a conceptual description of knowledge and knowledge development, however. An approach for knowledge access and development in firms is given by Boisot [6]. Here, development scenarios of knowledge in the Information Space are provided. For the conception part of knowledge development, there exists the well-known approach by Nonaka/Takeuchi [14], which is built on the distinction between tacit and explicit knowledge and on four knowledge conversions between the knowledge types (SECI-model). Approaches for knowledge transfer are surveyed in [13]. Concepts for organizational learning, which is closely related to knowledge management, are given by Argyris and Schön [4, 5] and by Senge [17]. The latter refers to system thinking as very important fifth discipline of organizational learning. In [3] a new conception of organizational learning based on knowledge dynamics is presented.

For intellectual capital, which is a more strategic view on knowledge in a company, see [19] for an approach towards an ontology for this domain.

In this paper, we propose a conception towards an ontology for enterprise knowledge management. To this end, we first summon up the tasks of knowledge management in an enterprise from a process-oriented point of view. Important items are knowledge processes, knowledge management processes, knowledge flows, and organizational learning. Second, we explain a conception of knowledge itself and of knowledge dynamics.

Based on this, we present a new conception for a formalized model for enterprise knowledge management. It consists of a layered set of ontologies. This set includes ontologies for knowledge and knowledge dynamics, for human interaction, for management, and for the whole enterprise. They together will support the mentioned processes related to knowledge management.

One of the basic constituents of this model is presented in detail as a semantic implementation of the conception of knowledge and knowledge dynamics, namely a corresponding ontology and rule system. Other constituents of the model have yet to be developed.

The structure of the paper is as follows. After an introduction, section II provides an outline of knowledge management and its tasks from a process-oriented point of view. This reflects knowledge processes, knowledge management processes, knowledge flows and organizational learning. Section III shortly presents the conception of knowledge and of knowledge dynamics. Then, section IV introduces the overall semantic-based concept as a layered set of ontologies with special recognition of the processes and tasks identified in section II. Section V describes the developed ontology for knowledge and knowledge development with the corresponding rule system. A summary and outlook section will conclude the paper.

 ISBN: 978-1-61208-175-5

II. OVERVIEW ON TASKS AND PROCESSES OF KNOWLEDGE MANAGEMENT

In this section, an overall view on the tasks and processes of knowledge management is given from a process-oriented point of view. We describe knowledge processes, knowledge management processes and knowledge flows as essentials parts of knowledge management. In addition organizational learning is shortly explained, which is closely related to knowledge management.

The extended knowledge cycle was originally introduced by Probst [16] as far as the outside cycle is concerned. Lehner [12] in addition introduced the correspondence to knowledge-intensive business processes and the knowledge flows. This again has been rearranged and changed by the author to the version as given in Figure 1.

As basic notion we have knowledge processes (depicted as yellow activities in Figure 1), which compose a whole knowledge cycle from identification, acquisition, structuring (constructing, combining, representing), storage, distribution (communication), usage until keeping and preservation. They may be grouped into four areas: preservation of new and existing knowledge, generation of new knowledge, making available knowledge, and using knowledge. These groups are indicated by the dotted rectangles in Figure 1. Two additional special knowledge processes (the blue arrows in Figure 1) are meta-level processes and close the overall cycle by goal-setting, knowledge evaluation and the feedback.

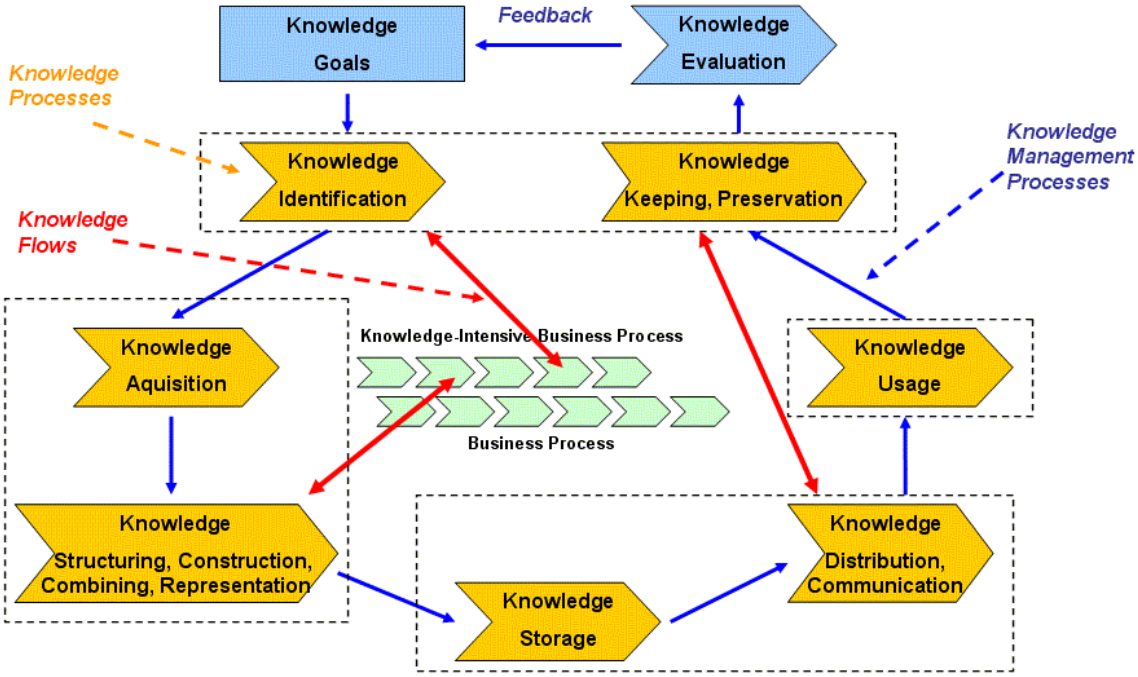


Figure 1. Tasks and Processes of Knowledge Management (Sources: Ammann, reworked from Probst [16] and Lehner [12])

Knowledge Management Processes keep the knowledge cycle going. Knowledge goals are set and drive the knowledge cycle until an evaluation. In general the blue arrows in Figure 1 represent knowledge management processes. For example, a knowledge management process takes care inside the above-mentioned knowledge process group “making knowledge available”, that employees are encouraged to communicate knowledge. The final feedback in the cycle is an important knowledge management process. Here gained knowledge is compared against the original goal and possibly a new cycle with a new or changed goal is initiated.

In our process-oriented view, business processes of the company, especially the knowledge-intensive ones, relate to knowledge processes. For example, in an earlier activity of the business process the need for new or re-combined knowledge is becoming clear, while in a later phase this knowledge is communicated to certain employees. This relation is provided by knowledge flows. In addition, knowledge flows can also interrelate different knowledge processes, as shown in Figure 1 between the knowledge distribution and knowledge preservation processes.

Organizational Learning is closely related to knowledge management. This resembles the classic triad composed of knowledge, learning, and storage. The latter one can be provided by the organizational memory. Organizational learning has been described with the help of single-loop, double-loop, and deutero learning, see [4, 5]. A novel

 ISBN: 978-1-61208-175-5

approach to build those organizational learning cycles on top of knowledge dynamics is given in [3]. See the following section for details on this knowledge and knowledge dynamics conception.

III. A CONCEPTION OF KNOWLEDGE AND KNOWLEDGE DYNAMICS

In this section, a conception of knowledge and knowledge dynamics in a company is shortly described. More details of this conception are given in [2].

A. Knowledge Conception

We provide a conception of knowledge with types, kinds and qualities as three dimensions. As our base notion, knowledge is understood as justified true belief (in the propositional kind), which is (normally) bound to the human being, with a dimension of purpose and intent, identifying patterns in its validity scope, brought to bear in action and with a generative capability of new information, see [1, 10, and 12]. It is a perspective of "knowledge-in-use" [8] because of the importance for its utilization in companies and for knowledge management.

The type dimension is the most important for knowledge management in a company. It categorizes knowledge according to its presence and availability. Is it only available to the owning human being, or can it be communicated, applied or transferred to the outside, or is it externally available in the company's organizational memory? It is crucial for the purposes of the company, and hence a main goal of knowledge management activities, to make as much as possible knowledge available, i.e. let it be converted from internal to more external types.

Our conception for the type dimension of knowledge follows a distinction between the internal and external knowledge types, seen from the perspective of the human being. As third and intermediary type, explicit knowledge is seen as an interface for human interaction and for the purpose of knowledge externalization, the latter one ending up in external knowledge. Internal (or implicit) knowledge is bound to the human being. It can be further divided into conscious, latent and tacit knowledge, where those subtypes do partly overlap with each other; see [10]. It is all that, what a person has "in its brain" due to experience, history, activities and learning. Explicit knowledge is "made explicit" to the outside world, e.g., through spoken language, but is still bound to the human being. External knowledge finally is detached from the human being and may be kept in appropriate storage media as part of the organizational memory.

In the second dimension of knowledge, four kinds of knowledge are distinguished: propositional, procedural and strategic knowledge, and familiarity, resembling to a certain degree the type dimension in [8]. Propositional knowledge is knowledge about content, facts in a domain, semantic interrelationship and theories. Experience, practical knowledge and the knowledge on "how-to-do" constitute procedural knowledge. Strategic knowledge is meta-cognitive knowledge on optimal strategies for structuring a problem-solving approach. Finally, familiarity is acquaintance with certain situations and environments; it also resembles aspects of situational knowledge, i.e. knowledge about situations, which typically appear in particular domains.

The quality dimension introduces five characteristics of knowledge with an appropriate qualifying and is independent of the kind dimension: level, structure, automation, and generality. See [2, 8] for more details.

This knowledge conception can be visually represented by a knowledge cube as shown in Figure 2.

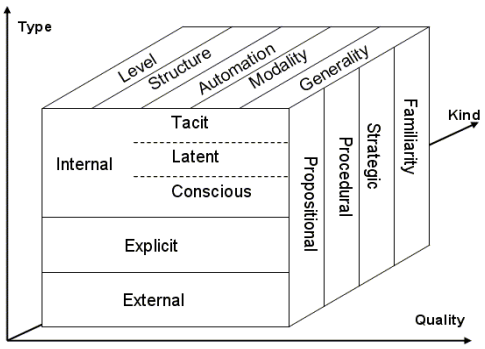


Figure 2. The knowledge cube

B. Knowledge Dynamics

Knowledge conversions, i.e. the transitions between the different knowledge types, kind and qualities between or within humans are responsible to a high degree for knowledge development in an organization. These conversions are the building blocks to model knowledge dynamics, i.e., all of acquisition, conversion, transfer, development and usage of knowledge, in an enterprise.

Five basic knowledge conversions in the type dimension are distinguished here: socialization, explicitation, externalization, internalization and combination. Basic conversion means, that exactly one source knowledge asset is converted into exactly one destination knowledge asset and exactly one knowledge dimension (i.e. the type dimension in this case) is changed.

Socialization converts tacit knowledge of a person into tacit knowledge of another person. This may succeed by exchange of experience or in a learning-by-doing situation. Explicitation is the internal process of a person, to make internal knowledge of the latent or conscious type explicit, e.g. by articulation and formulation (in the conscious case) or by using metaphors, analogies and models (in the latent case). Externalization converts from explicit knowledge to external knowledge or information and leads to detached

 ISBN: 978-1-61208-175-5

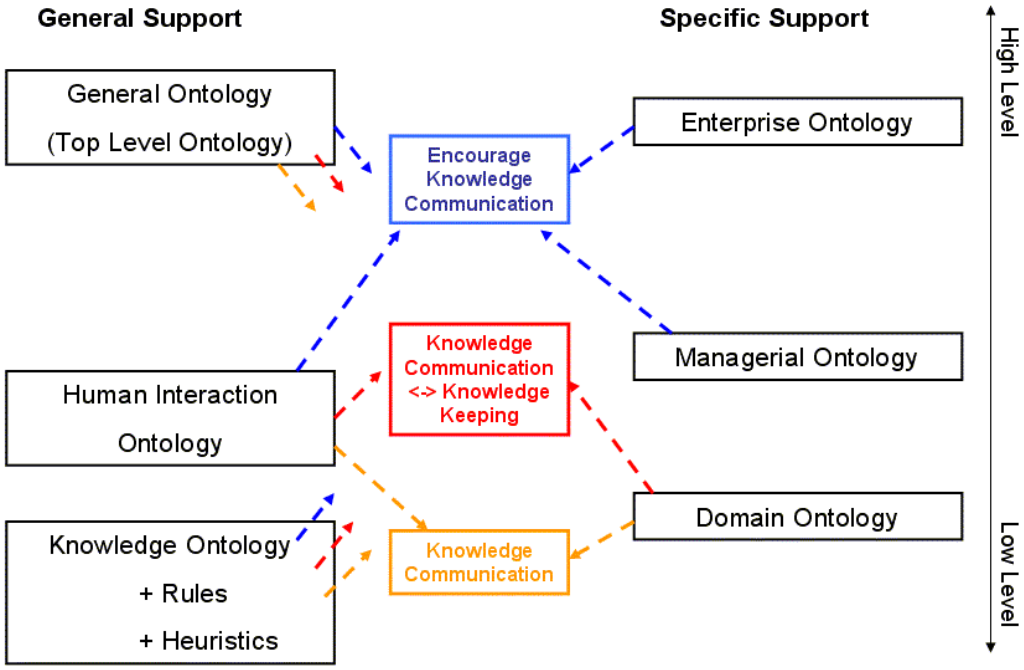


Figure 3. Layered set of ontologies with rule systems

knowledge as seen from the perspective of the human being, which can be kept in organizational memory systems. Internalization converts either external or explicit knowledge into internal knowledge of the conscious or latent types. It leads to an integration of experiences and competences in your own mental model. Finally, combination combines existing explicit or external knowledge in new forms.

Basic knowledge conversions in the kind dimension of knowledge do not occur. Those in the quality dimension are mostly knowledge developments aiming at quality improvement. Examples include basic conversions changing the overview, structure and automation quality, respectively.

More complex conversions can be easily gained by building on this set. They consist of n-to-m-conversions and include information assets in addition. General knowledge conversions convert several source assets (possibly of different types, kinds and quality) to several destination assets (also possibly different in their knowledge dimensions). In addition, information assets are considered as possible contributing or generated parts of general knowledge conversions.

IV. OVERALL SEMANTIC CONCEPT OF KNOWLEDGE MANAGEMENT

Having provided the tasks and processes of knowledge management in section II and a conception of knowledge and knowledge dynamics in section III, we now proceed with the introduction of an overall concept for semantic support of knowledge management. This can be viewed as a step towards an ontology (or a set of ontologies) for knowledge management.

Figure 3 depicts this conception of a layered set of ontologies and gives a example, how knowledge processes, knowledge management processes, and the knowledge flows are supported by the various ontologies in this conception. We propose a hierarchical structure, which is also divided in a general and a specific part. At the general support side, we start with an ontology of knowledge and knowledge dynamics at the bottom layer. The Knowledge Ontology as described in the following section V implements the corresponding conception as introduced in section III. It is complemented by a set of rules and (in the future) of heuristics, which enhance the support for reasoning in incomplete knowledge application scenarios. An incomplete scenario consists of one or more general knowledge conversions, where one or more places (source or destination knowledge objects or conversions themselves) are not known. They may be implied by an application of an appropriate rule or a heuristics. While rules support the proper handling of knowledge conversions and transfers, heuristics will be needed for those cases of knowledge dynamics, where no unique resolution of source and destination knowledge assets in complex knowledge conversions is possible with rules. The following section V will describe the Knowledge Ontology and the corresponding rule system.

Built on top of the Knowledge Ontology a Human Interaction Ontology conceptualizes human-to-human interactions. The knowledge and knowledge dynamics

 ISBN: 978-1-61208-175-5

support is utilized here, based on the observation that human-to-human interaction always comes along with knowledge transfers (conversions). To state is differently, human-to-human interaction can be modeled by appropriate general knowledge conversions between people. As top layer on the general side, a top level ontology will provide general concepts like time, locations, and so on.

On the specific support side, one or more Domain Ontologies reflect the domains of interest in the enterprise. On top of it, a Managerial Ontology provides management conceptions related to knowledge management. This again is utilized on the next layer by an Enterprise Ontology, which conceptualizes the whole (specific) enterprise.

Figure 3 gives an example how the knowledge processes, knowledge management processes, and the knowledge flows are supported by the various ontologies in this conception. The same color code is used in Figure 3 as in Figure 1. Each type of processes is supported by the Knowledge Ontology and the General Ontology on the general side. A knowledge process like "knowledge communication" utilizes the Human Interaction Ontology and the appropriate specific Domain Ontology in addition. The same kind of support can be observed for knowledge flows, as can be seen for the flow from "knowledge communication" to "knowledge keeping" in Figure 3. Finally knowledge management processes like "Encourage Knowledge Communication" will take hold of the Human Interaction Ontology from the general side and the Managerial and Enterprise Ontologies from the specific side.

V. THE KNOWLEDGE ONTOLOGY

In this section we present the Knowledge Ontology, which implements the conception of knowledge and knowledge dynamics as described in Section III. It is one of the building blocks in the set of ontologies as described in section IV. Here we describe the ontology, restrictions and reasoning, and rules. For more details, see [2].

The ontology (as visually shown in Figure 4) is divided in four core concepts: *Knowledge*, *Information*, *Knowledge_Conversion* and *Knowledge_Dimension*. The three different knowledge dimensions are represented as: *Type_Dimension*, *Kind_Dimension* and *Quality-Dimension*. *Knowledge* is defined according to these dimensions. Properties are used to model the relationships between *Knowledge* and *Dimensions*: *hasType*, *hasKind* and *hasQuality*. For example, *Explicit_Knowledge* is defined as every piece of knowledge, which is related to the instance *Explicit_Type* via the *hasType* property. In the same way, *Knowledge* in general must be related to every quality sub-dimension through the *hasQuality* property.

Two properties have been defined to model the knowledge conversions: *hasSource* and *hasDestination*, with knowledge conversions as ranges, and pieces of knowledge and information as domains.

A General Conversion is modeled through the *Knowledge Conversion* concept, and its only restriction is the fact that it must have at least one source asset and one destination asset. *Basic Conversions* are more specific, in the sense that

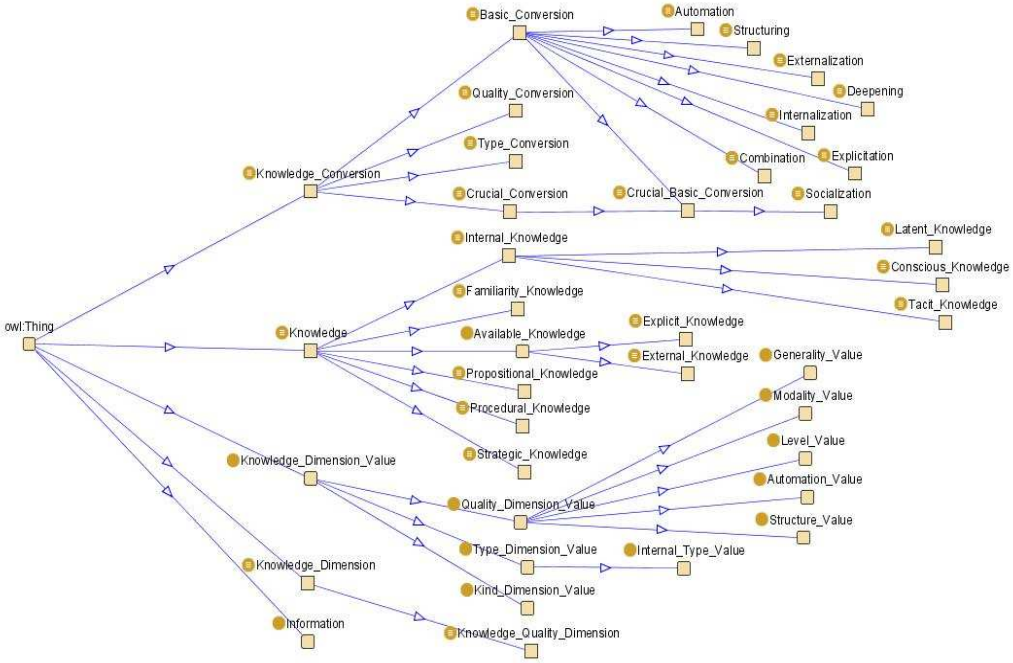


Figure 4. Knowledge ontology hierarchy

 ISBN: 978-1-61208-175-5

they have only one source and only one destination. The concept *Crucial_Conversion* gathers those conversions that contribute to the goal of making the knowledge available for the company.

Basic reasoning is based on subsumption mechanisms that deal with the ontology hierarchy. However, ontologies can contain more complex elements to enable advanced reasoning. In this way, the Knowledge Ontology has been extended with OWL restrictions to enable new ways of generating interesting new knowledge.

Ontology restrictions allow us to infer new characteristics of a given concept or instance. However, in some cases we could require to generate new instances in the ontology depending on certain situations. In these cases rules have been used, so the Knowledge Ontology will be able to infer all the possible conversions given some pieces of knowledge. SWRL [18] rules have been defined and the Jess rule engine [11] has been used. One rule will create basic conversions with all the possible source-destination pairs, and then, the same engine will characterize these conversions, inferring the changing dimension for each case. Six further rules have been established to infer the changing dimensions of each of the new discovered conversions: one for the type dimension and five for the quality ones. For example, the rule for the type dimension is as follows:

```
Knowledge(?k1) ^ Knowledge(?k2) ^
hasTypeValue(?k1, ?v1)  ^ hasTypeValue(?k2, ?v2) ^
differentFrom(?v1, ?v2) ^ Knowledge_Conversion(?c1) ^
hasSource(?c, ?k1) ^ hasDestination(?c, ?k2)
 →
hasChangingDimension(?c,
                Knowledge_Type_Dimension)
```

This development has already opened the path, to solve open questions in application scenarios for knowledge development. With the help of representations, these scenarios can be mapped to general knowledge conversions, which are subject to rule processing in relation to the Knowledge Ontology. A final interpretation step leads back to the solved scenario. See [2] for examples of some application scenarios solved with this method.

VI. SUMMARY AND OUTLOOK

An overall semantic conception for enterprise knowledge management has been given in this paper. It consists of a layered set of ontologies of the important and relevant sub-domain of this domain. This conception was motivated by the observation of tasks of processes of knowledge management, i.e. knowledge processes, knowledge management processes, and knowledge flows.

One of the basic constituents of this conception, namely the knowledge ontology together with reasoning support and a rule system already exists and has been described in this paper.

Future work includes the development of the other ontologies in our layered set of ontologies on the one side and an implementation of knowledge processes, knowledge management processes, knowledge flows and organizational learning cycles based on the set of ontologies on the other side.

REFERENCES

[1] Ammann, E.: “A Meta-Model for Knowledge Management”, in: 5th Int. Conf. on Intellectual Capital and Knowledge Management (ICICKM), New York 2008, pp 37-44.

[2] Ammann, E., Ruiz-Montiel, M.,Navas-Delgado, I., Aldana-Montes, J.F.: “Knowledge Development Conception and its Implementation: Knowledge Ontology, Rule System and Application Scenarios”, in: Proceedings of the 2nd Int. Conf. on Advanced Cognitive Technologies and Applications (COGNITIVE 2010), Lisbon, Portugal, 2010, pp. 60-65.

[3] Ammann, E.: “Knowledge Dynamics and Organisational Learning Cycles”, in: Proc. of the 12th European Conference on Knowledge Management 2011, Passau, Germany, 2011, pp.10-19.

[4] Argyris, C. and Schön, D.A.: Organizational Learning: a Theory of Action Perspective, Addison-Wesley, Reading, Massachusetts 1978.

[5] Argyris, C. and Schön, D.A.: Organizational Learning II – Theory, Method, and Practice, Addison-Wesley, Reading, Massachusetts 1996.

[6] Boisot, M.H. Knowledge Assets, Oxford Univ. Press 1999.

[7] Bubenko, J.A., Jr., Brash, D., and Stirna, J.: EKD User Guide, Dept. of Computer and System Science, KTH and Stockholm University, Elektrum 212, S-16440, Sweden 1998.

[8] De Jong, T., Fergusson-Hessler, M.G.M. “Types and Qualities of Knowledge”, Educational Psychologist, 31(2), 1996, pp.105-113.

[9] EKD – Enterprise Knowledge Development, ekd.dsv.su.se/home.html.

[10] Hasler Rumois, U. Studienbuch Wissensmanagement (in German), UTB orell fuessli, Zürich 2007.

[11] Jess Rule Engine, http://www.jessrules.com.

[12] Lehner, F.: Wissensmanagement (in German), 3rd ed., Hanser, München 2010.

[13] Ling, L.H.: “From Shannon-Weaver to Boisot: A Review on the Research of Knowledge Transfer”, in: Proc. of Wireless Communications, Networking and Mobile Computing (WiCom 2007), 2007, pp. 5439-5442,

[14] Nonaka, I. and Takeuchi, H.: The Knowledge-Creating Company, Oxford University Press, London 1995.

[15] OWL Web Ontology Language Reference, http://www.w3.org/standards/history/owl-ref.

[16] Probst, G., Raab, St., Romhardt, K.: Wissen managen (in German)”, Gabler, 6th ed. 2010.

[17] Senge, P.: The Fifth Discipline, Currency Doubleday, New York 1994.

[18] SWRL: A Semantic Web Rule Language Combining OWL and RuleML, http://www.w3.org/Submission/SWRL/ .

[19] Vlismas, O. and Venieris, G.: “Towards an Ontology for the Intellectual Capital Domain”, in: Journal of Intellectual Capital, Vol.12, No. 1, 2011, pp. 75-110.

 ISBN: 978-1-61208-175-5

SIGA3D: A Semantic BIM Extension to Represent Urban Environnement

An ontology-based management of level of details

Clément Mignard[1], Gilles Gesquière[2], Christophe Nicolle[3]
[1]Active3D, 2 rue René Char, BP 66 606 21066 Dijon Cedex, France
[2] Aix-Marseille Université, LSIS - UMR 6168, IUT BP 90178 - 13637 ARLES, France
3 LE2I – UMR 5158, IUT Dijon-Auxerre, Université de Bourgogne, BP 47870, 21078 Dijon Cedex, France
[1]c.mignard@active3d.net, [2]gilles.gesquiere@lsis.org, [3]cnicolle@u-bourgogne.fr

***Abstract*— This paper presents a new architecture dedicated to the management of buildings and urban objects through a 3D digital mockup. We focus on the ontology-based framework of this architecture, and the semantic LoD (Level of Detail) mechanism defined to build dynamically the 3D scene from a set of heterogeneous information systems. This project is developed into an industrial web platform which manages more than 100 million square meters of buildings.**

Keywords- Interoperability; Semantic Heteogeneity; ontology; used profil; Building Information Modelling; Geographic Information Systems

I. INTRODUCTION

Today, at a time when environmental issues are becoming more insistent, ways to control costs in the management and development of a territory are increasingly sought. This may involve the facility management of a set of buildings that one wishes to identify and observe to limit the costs of maintenance or the creation of new entities to anticipate the ecological and economic impacts. These goals require a lot of heterogeneous information on assets to manage, at several moments of their life cycle. This unification is an expensive process which is not always adapted to the trends of the trade or the market. The global information system becomes quickly obsolete and unsuited regarding the data model evolutions and improvements. In order to unify and centralize the management of real estate, urban and extra urban, it is necessary to develop a new form of collaborative architecture. This architecture makes it possible to combine in a homogeneous environment a set of heterogeneous information from diverse information systems such as those from the Building Information Modeling (BIM) domain and the Geographical Information Systems (GIS) domain.

The term BIM has been coined recently to demarcate the next generation of Information Technologies (IT) and Computer-Aided Design (CAD) for buildings, which focus on drawing production. BIM is the process of generating, storing, managing, exchanging and sharing building information in an interoperable and reusable way. A BIM system is a tool that enables users to integrate and reuse building information and domain knowledge throughout the building life cycle [2].

GIS are becoming a part of mainstream business and management operations around the world in organizations both in public and private sectors. The term GIS refers to any system that captures, stores, analyzes, manages, and presents data that are linked to at least one location.

Since 2008, we develop a collaborative web platform dedicated to urban facility management. This approach is based on a semantic architecture using ontology evolution mechanisms. The content of this ontology can be displayed in a real time 3D viewer we have developed. This one allows the management of a large number of objects in scenes and the management of geocoding objects by implementing a mechanism of geometric Levels of Details (LoD). In our architecture, we introduced also a semantic multi-representation mechanism (i.e. several semantic definitions of a concept depending of local contexts).

This approach of multi-representation adds to the traditional principle of LoD the notion of Contextual LoD (C-LoD). A C-LoD is a geometric representation of an object which is selected according to semantic criteria and not only displayed depending on the distance between the view point and the object as it is usually the case for LoD. The criteria may depend on user (we defined a profile in which we can find various information like the business process to which he is attached), external criteria as day/night or weather, or even of the object itself (intrinsic properties such as material, temperature, etc.). The semantic management drives streaming processes, which extract the knowledge and 3D representation of urban objects from a relational database. Moreover, all the technologies used to build our framework architecture attempt to be as compatible as possible with the standards in use in the semantic, geospatial and BIM worlds. This allows us to bridge the gap of interoperability meet at different levels when working with several data sources coming from several domains.

II. SEMANTIC REPRESENTATION OF URBAN ENVIRONMENT

Our proposal is based on a semantic architecture articulated in 6 levels (Fig. 1). The import/export level is dedicated to the parsing of various file formats required to model a complete urban environment from different sources (GIS/BIM). This can be done from local files or Web Services. The Data Model Framework (DMF) level makes it possible the combination of geometrical data and semantics.

 ISBN: 978-1-61208-175-5

The level "Contextual View" associates user profiles and business rules to build C-LoDs. The connection level is mainly dedicated to the streaming process between the databases and the interface. The interface level displays the urban environment into a 3D digital mockup coupled with a semantic tree of urban elements.

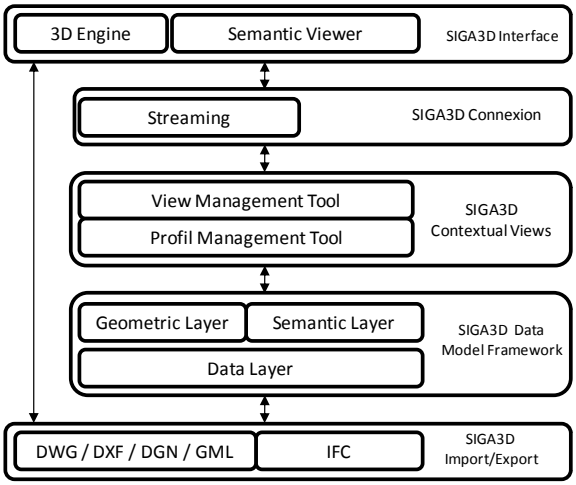


Fig. 1. SIGA3D Architecture.

The innovative feature of this architecture is mainly contained in the DMF level and Contextual Views level. These levels are the base of our semantic C-LoD proposal. The DMF level is made of graphs representing the ontology, allowing the context management and versioning of the data (through CMF for Context Model Framework which matches with the Contextual Views layer of the Figure 1). Graphs operators are defined to facilitate the implementation of changes in conceptualization. Information about reference systems for space and time (Coordinate Reference System (CRS) and TimeZone) are also managed in this part. The other part, DMF, defines a unified syntax-based knowledge representation based on the languages OWL, RDFS, and rules RuleML, SWRL and described in this document in an expressive way with description logic. DMF also contains operators for the management of space and time and the definition of local contexts that allow us to conduct a multi-representation of data. The goal of this part is to provide models used in an inference engine to infer and to check the data modeled by the C-DMF (Context-Data Model Framework which include CMF and DMF) modeling operators.

III. SIGA3D DATA MODEL FRAMEWORK

The Data Model Framework is made of operators to construct urban data models. These operators allow the description of classes and properties that can be used to define complex concepts using operators of intersection, union, involvement, etc.

dmf:Class defines a class.
dmf:Property defines a property.
dmf:Var defines variables used in the logical formulas.
dmf:Predu defines unary predicates.
dmf:Predb defines binary predicates.
dmf:Equiv defines two predicates as equivalent.
dmf:And defines the intersection.
dmf:Not defines the negation.
dmf:Or defines the union.
dmf:OrX defines the exclusive disjunction.
dmf:Diff defines the difference.
dmf:Imp defines the implication. It is used to represent various operators like sub-property, restriction, transitivity, symmetry, functional property, etc.
dmf:spatialEntity defines a geometric representation. This operator refers to a geometrical representation of the object with IFC or CityGML standard.
dmf:temporalEntity defines an instant or an interval of time.

The spatial data and especially georeferenced coordinates do not make sense without the knowledge of the coordinate reference system. This information appears in the upper layer of our architecture that manages the context of model graph, to unify the management of coordinates. The same kind of information is provided for time, with the management of Time zones.

The management of local contexts, which allows multi-representation, is done in this part by defining new stamped operators (based on the mechanism described in the part V of this article), corresponding to the DMF operators defined above. For example, the script 1 defines three local contexts, *designer*, *structureEngineer* and *March*.

```
<dmf:Class rdf:ID='Profession'/>
  <Profession rdf:ID='designer'/>
  <Profession rdf:ID='structureEngineer'/>
<dmf:temporalEntity rdf:ID='achievementDate'/>
<dmf:property rdf:ID='unitType'/>
  <Day rdf:ID='March'>
    <unitType rdf:resource='#unitMonth/>
   </Day>
```

Script 1. Definition of local contexts.

We can then define several properties and a spatial representation for a class '*buildingPlan*' which depends of the user. In the script 2, the contextual operators *dmf:[c_1, …,c_n]Class*, *dmf:[c_1, …,c_n]property* and *dmf:[c_1, …,c_n] spatialEntity* are used.

```
<dmf:Class rdf:ID='BuildingPlan'/>
<dmf:[designer]property rdf:ID='line_thick'/>
<dmf:[structureEngineer]property
       rdf:ID='wall_material'/>
<dmf:[designer]property rdf:ID='contains_plan'/>
<dmf:[designer,structureEngineer]property
             rdf:ID='contains_plan'/>
<dmf:spatialEntity rdf:ID='the_plan'/>
<dmf:[designer]property rdf:ID='3D_plan'/>
```

 ISBN: 978-1-61208-175-5

```
<dmf:[designer,structureEngineer]property
rdf:ID='2D_plan'/>
  <the_plan rdf:ID='plan_of_building_1'>
   <url_2D_plan
      rdf:resource='/building/1/plan/plan2D.dwg'/>
   <url_3D_plan
      rdf:resource='/building/1/plan/plan3D.ifc'/>
  </the_plan>
<dmf:[designer,March]Class
       rdf:ID='Plan_availability'/>
    <BuildingPlan rdf:ID='building_plan_1'>
    <line_thick rdf:dataType='&xsd;float'>10
    </line_thick>
    <wall_material rdf:dataType='&xsd;float'>wood
    </wall_material>
    <contains_plan rdf:resource='the_plan'/>
  </BuildingPlan>
```

Script 2. Use of contextual operators.

This example describes an object, *BuildingPlan*, which has several properties. For a designer, the *BuildingPlan* is defined with a *line_thick* and a *plan* contains two representations. The same object is defined differently for a structure engineer, with the material of walls, *wall_material*, and an attached plan with only one 2D representation. The figure 2 shows another example of multi-representation on a building storey. On the left part we have a structural view of the building according to the bricklayer context, and on the other side we can see a woodwork view (flooring, windows, doors and stairs) according to the joiner context (right part).

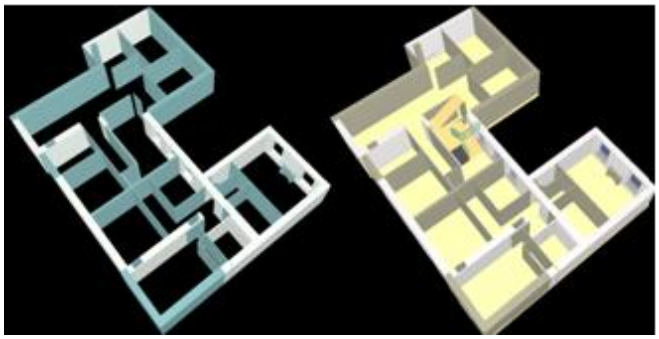


Fig. 2. Example of semantic multi-representation of a building.

IV. SIGA3D CONTEXT MODEL FRAMEWORK

This part of our architecture is composed of three main blocks. The first block sets the context for each graph of DMF. The second block defines a set of graph operators to facilitate the writing of information and limit the redundancy of data in the context management. Then the third block defines a set of operators on graphs to describe more accurately the geographical information by defining spatio-temporal relations between different data models of DMF. Context management in this architecture is done by defining a special graph called *SystemGraph*. A *SystemGraph* defined the context for a graph or a set of graphs using operators. These operators can be applied on graphs defined in the second block of the CMF. The use of these operators can simplify the management of the evolution of knowledge of the model. Indeed, rather than storing for each modification of the model a new version of the complete graph, the CMF layer stores the modification as operations on graphs. The *SystemGraph* can be described using the following operators:

- *cdmf:graph* connects graph and data. These data are described according to the data model. They can be a combination between other graphs using the CMF graph operators *AddGraph* (union of graphs), *RemoveGraph*, *InterGraph*, *CompInterGraph* and *MapGraph*. These operators allow us to improve the modification tracking of the ontology by limiting the size of the graphs and their reusability.
- *cdmf:of* represents the context. This property defines a list of resources representing the access context.
- *cdmf:model* refers to the data model which is used. This data model defines elements which will appear in the graph.
- *cdmf:action* defines user's rights to access the data (read/write/remove). If no action is defined in the system Graph, which means that only the visualization of the data is allowed.
- *cdmf:synchronizationGraph* defines a list of graphs linked with the element *cdmf:graph* by all kind of spatial and temporal relationship.
- *cdmf:reference_frame* defines the TimeZone and the CRS used for the data model associated to the *SystemGraph*. These values are valid for all data of associated graphs. This means that if original data sources are not defined in the same CRS, a transformation of coordinates has to be done before using the data.

The spatio-temporal synchronization is not a common graph operator and is very specific to the description of geographical information. It allows defining the validity of a model by describing relationships with other models. It can be used in case of model evolution to assure the consistency of the global model. For instance, if we define a building model and an electric power network model, it is possible to describe a topological relation between the two models to say they are spatially connected. Then, if one of the models is modified, for example, to move the building in the case of a bad georeferencing, the other model has to be modified to keep the spatial connection relation consistent.

V. PRINCIPLE OF SEMANTIC MULTI-REPRESENTATION

The principle of multi-representation can consist to display different maps of different scales for a same place, or to simplify the geometry of an object depending of geometric criteria such as distance or size. This is the well-known mechanism of LoD in GIS. To this geometric

 ISBN: 978-1-61208-175-5

definition of multi-representation, we propose to add a semantic dimension. This semantic multi-representation allows a user to display information in a form that suit him (contextual view), or to make a control on access of the modeling data. The combination of these two types of multi-representation is an innovative aspect of our approach. It gives the new concept of C-LoD, representations that would be displayed according to semantic criteria.

To implement this new mechanism, it is needed to have a formalization of the multi-representation system in a semantic way. The works based on the MADS approach by [8] and later by [1] define a multi-representation formalism in ontologies. This approach is based on a stamping mechanism of the representations. In our architecture, stamps can be defined with any element of the DMF layer and especially spatial and temporal elements. Moreover, stamps can be applied on every element and operators of the DMF layer, such as data, instance of types and values of attributes, meta-data, and definition of a type or an attribute of the schema. The local context mechanism of the building and urban modeling architecture is based on this formal approach. Associating to the concept of local context, it is used to define contextual operators to model these contexts. A part of these operators is already defined in the BIM part with the possibility to build contextual view.

The next step required is the definition of operators for GIS domain. Thus, the local context can be also defined with spatiotemporal operators to describe the objects depending on space and time, an important dimension of GIS.

VI. DISCUSSION

Interoperability may be defined as "the ability of two or more systems or components to exchange information and to use the information that has been exchanged" [4]. Systems that can exchange data are syntactically interoperable: they share a common structure, with agreed-upon data formats and communication protocol. Syntactical interoperability is a prerequisite for further interoperability. The ability for systems to interpret automatically the exchanged information is known as semantic interoperability. The same meaning can be derived from the data at both ends. This implies that the systems share a common information model, where each element of the model is precisely defined. In a world where software vendors have implemented products tailored to the needs of specific communities and/or customers, standardization is the most efficient and global solution to interoperability problems [10]. Several organizations, industry consortiums and communities are involved in standards development activities related to urban matters:

- ISO/TC 211 (International Organisation for Standardization / Technical Committee 211, http://www.isotc211.org/) - Geographic Information and Geomatics is responsible of standards for geospatial information;
- Open Geospatial Consortium (OGC, http://opengeospatial.org/) focuses on standards for geospatial services;
- The buildingSMART alliance (formerly IAI for International Alliance for Interoperability, http://www.buildingsmartalliance.org) focuses on developing standards for the construction and facility management industries;
- Web3D Consortium (http://www.web3d.org/) is concerned with standards for 3D data exchanged over the Internet;
- Khronos Group (http://www.khronos.org/) creates open standards for the authoring and acceleration of parallel computing and graphics media;
- ISO/TC 204 – Intelligent transport systems standardizes information, communication and control systems in the field of surface transportation.

The use of standards that allow joint exploitation and combination of various geospatial and CAD data is a requirement for developing interoperable systems and is an increasing demand from user communities. In our case, to build the Contextual LoD, we have to share 3D models and its relationship with semantics. User communities can take advantage of this framework of standards to develop application schemas that follow the rules and reuse the components defined in the abstract standards. An XML Schema encoding following the GML grammar can then be derived from the application schema and serve as the basis for data exchange. This approach was followed during the development of CityGML and INSPIRES data specifications.

ISO/TC 211 has started to standardize different thematic aspects of geospatial information. Several standardized conceptual schemas have been defined, in accordance with ISO 19109. The following standards are relevant to urban space modelling:

- ISO 19144-2 - Classification systems - Part 2: Land Cover Meta Language (LCML) defines a meta language for expressing land cover classifications. Land cover classifications can be used to distinguish built-up areas from non-urban zones.
- ISO 19152 – Land Administration Domain Model (LADM) is a standardized conceptual schema for cadastre data. Land administration data can also play an important role in urban models.
- ISO/TC 204 has developed ISO 14825 - Geographic Data Files (GDF) as a conceptual and logical data model and exchange format for geographic databases for transportation applications. GDF has a strong focus on road transportation information.

 ISBN: 978-1-61208-175-5

Other organizations have developed and maintain standards for urban and building models. The Industry Foundation Classes (IFC), defined by buildingSMART, is a BIM data schema covering a wide range of information elements required by software applications throughout the life cycle of a building. IFC now contains more than 700 classes enabling the exchange of building design, construction and maintenance data [7]. IFC 2.3 was adopted as ISO/PAS 16739 in 2005. The next release of IFC, IFC4, will be published as the ISO standard ISO/IS16739 at the end of year 2011 and will feature an improved modeling of external spaces and better support for geographic coordinate reference systems.

OGC published CityGML 1.0 in 2008. CityGML specifies a standardized application schema for 3D city models, from which a GML 3.1.1 encoding is derived. CityGML is therefore, both a conceptual model and an encoding, enabling syntactic and semantic interoperability. Its key features [5] are:

1. Thematic modeling: the model covers a wide range of city objects, including but not limited to buildings, transportation facilities, water bodies, vegetation…
2. Modularization: each thematic model is packaged in a separate UML module.
3. Multi-scale modeling: CityGML supports five levels of details (LoD). This mechanism facilitates the integration of 2D (at LoD0) and 3D datasets at distinct scales representing the same real-world entities. The same feature can be represented with different geometries at each scale. CityGML also provides an aggregation and decomposition association between objects that can be used to indicate that an object at a lower LoD has been decomposed into two or more objects at a higher LoD. They are defined as follows:
   - LoD0: regional view. An ortho-image or a map may be draped over a Digital Terrain Model, together with regional LandUse, water bodies and transportation information;
   - LoD1: city view. Buildings are modeled as flat-roofed blocks;
   - LoD2: city district, project view. Buildings are modeled with distinct roof structures and semantically-classified boundary surfaces. Vegetation objects, city furniture and more detailed transportation objects may also be modeled.
   - LoD3: architectural models (outside), landmark. Detailed wall and roof structures, balconies, bay and projection structures are modeled, as well as high-resolution textures, complex vegetation and transportation objects.
   - LoD4: architectural models (inside). Interior structures are modeled.
4. External references: objects in external databases may be referenced from the building or city object to which they correspond. They can be used to propagate updates from the source database to the 3D city object. They also help in linking different information models, while keeping them separate, as each has its own purpose.
5. Application Domain Extension (ADE) is a key mechanism of CityGML. Users can formally extend the base UML model with domain-specific information, e.g. an extension for utility networks or describing noise rates on city objects, and encode it in a XML Schema. Several ADEs have been developed for topics such as Noise (in relation with the European Noise Directive), Tunnels or Bridges. An ADE extending CityGML with more detailed semantics from the IFC standard is also being developed as the GeoBIM ADE [3].

CityGML's modularity, thematic structure, extensibility and external referencing mechanism sustain richer urban models integrating data from a variety of sources and enabling links with other application domains.

Semantic information must be taken into account according to 3D models. Transferring only geometry with the scene graph is not sufficient [6]. Transferring information between a server and a client application is not so easy. Using standards would be a good way. However, in our project, interactive exchange is needed and requires a semantic modeling of heterogeneous information using ontology [9].

VII. CONCLUSION

This paper presents an ongoing research on the definition of an Urban and Building Modeling Architecture. This paper focus on a new mechanism of LoD called Contextual LoD. It is the merge of classical geometric approach to define LoDs and two semantic multi-representations formalisms: the first part is based on contextual trees to define user profiles and business rules at the DMF level. The second part defines local contexts to allow multi-representation at a lower level, i.e. for each object of the model. The concept of C-LoD is designed to be integrated in an Urban Facilities Management (UFM) platform. It is an extension of the BIM concept for the management of urban objects. Our framework facilitates data maintenance (data migration, model evolution) during the life cycle of an urban environment and reduces the volume of data with specific graph operators. The urban approach also implies to manage precisely the spatial and temporal dimensions that have been considered in the definition of the C-LoD part. This approach is based on the CityGML 1.0 and IFC 2x3 standards. The implementation of the BIM part, including the making of data model and contextual views and profiles, as well as the 3D representation of building and urban objects with a LoD

 ISBN: 978-1-61208-175-5

management is already done. Our future works will be to achieve the implementation of our framework for the UFM platform, including the C-LoD management. These works are based on our previous works on Active3d and designed to be fully compatible with both standards: the one for geographic information (e.g. ISO/TC 211) and the second for the construction world (e.g. ISO16739).

REFERENCES

1. Benslimane D., Vangenot C., Roussey C.: A. Arara. Multi-representation in ontologies. Proceedings of 7th East-European Conference on Advances in Databases and Information Systems, ADBIS 2003, Dresden, Germany, September 3-6, (2003)
2. Campbell D. A.:Building Information Modeling: The Web3D Application for AEC, ACM Web3D, Perugia, Italy, (2007).
3. Döllner J., Hagedorn B: Integrating Urban GIS, CAD, and BIM Data By Service-Based Virtual 3D City Models. 26th Urban Data Management Symp., Stuttgart, Germany, (2007)
4. Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY: 1990
5. Kolbe T. H., Gröger G., Plümer L.: Citygml – interoperable access to 3d city models, In proceedings of the Int. Symposium on Geo-information for Disaster Management, pages 21–23, Delft, march 2005. Springer verlag (2005)
6. Kolbe, T. H.: Representing and Exchanging 3D City Models with CityGML, Lee, Jiyeong / Zlatanova, Sisi (Eds.), Proceedings of the 3rd Int.Workshop on 3D Geo-Information, Seoul, Korea. Lecture Notes in Geoinformation & Cartography, Springer Verlag, (2009)
7. Kuzminykh A., Ho_mann C: On validating STEP product data exchange, Computer-Aided Design, 40(2) :133- 138, (2008).
8. Parent C., Spaccapietra S., Zimanyi E. The MurMur project: Modeling and querying multi-representation spatio-temporal databases, Information Systems, 31 (8) (2006)
9. Vanlande, R., Cruz C., Nicolle, C.: IFC and Buildings Lifecycle Management",Journal of Automation in Construction, Elsevier, (2008)
10. Zhao P. and Di L., Geospatial Web Services : Advances in Information Interoperability, IGI Global, (2010)

 ISBN: 978-1-61208-175-5

Semantic Processing in IT Management

Andreas Textor, Fabian Meyer, Reinhold Kroeger
Distributed Systems Lab
RheinMain University of Applied Sciences
Unter den Eichen 5, D-65195 Wiesbaden, Germany
{*firstname.lastname*}*@hs-rm.de*

***Abstract*—In the domain of IT management, numerous models, protocols and tools have been developed. To achieve the long-term goal of comprehensive, highly automated IT management, the various sources of information need to be combined. As syntactic translation is often not sufficient, ontologies can be used to unambiguously and comprehensively model IT environments including management rules. In this paper, we present an approach that combines the domain model, rules, instance data (which represents real-world systems) into an ontology. Moreover, probabilistic knowledge of the domain is modeled using Bayesian networks and integrated into the ontology. A runtime system that aggregates data and merges it into the ontology, and then uses a reasoner to evaluate management rules, is described as part of the approach of the ongoing project.**

***Keywords*-ontology; IT management; Bayesian network**

I. Introduction

Knowledge bases grow in size and complexity in every domain. For this reason, in the domain of IT management, numerous models, protocols and tools have been developed. Notable models include the OSI network management model (also known as CMIP, the name of its protocol) and the still widely used simple network management protocol (SNMP). A more recent approach to specify a comprehensive IT management model is the Common Information Model (CIM, [1]), a widely recognized Distributed Management Task Force (DMTF) standard. The more complex an IT environment gets, the more important the capability becomes to automate as many tasks as possible. Both commercial and free management tools and frameworks exist that cover different parts of the required feature set for management tasks, but usually not only a single tool, but a set of tools is used. In order to achieve a unified view of the heterogenous integrated management models, mappings between different types of models can be defined. However, syntactic translations are often not sufficient, when the same concept is represented in a different way in different domains. This problem can be approached by using ontologies to clearly define the semantics.

Only when a comprehensive formal representation of the domain data exists, that is also capable of modeling rules, a largely automatic management becomes possible, because then not only structural, but also behavioural information is expressed in the model. To achieve such an automated management system, more prerequisites must be provided: A runtime system is required to import the corresponding domain model into the ontology and to evaluate the rules, based on up to date data from the managed system. Therefore, instance data must be acquired at runtime and added to the ontology, so that rules can be evaluated according to both model and instance data.

In certain cases, and especially in a domain as complex as IT management, the domain cannot be modeled solely using exact information, which might not be available. However, when relationships between entities are known and marked accordingly in the model, probabilistic evaluation is possible, where only incomplete data is available. To enable that, the ontology and the runtime system need to be extended accordingly.

The approach presented in this paper uses an OWL (Web Ontology Language, [2]) ontology to combine the domain model, instance data and rules defined in SWRL (Semantic Web Rule Language). To model entities and relationships of an IT environment, the CIM model was converted into an OWL ontology (the translation process is described in more detail in [3]). To model probabilistic knowledge, ontology elements are annotated so that a Bayesian network can be partially derived at runtime. Bayesian networks are a probabilistic model to specify causal dependencies between random variables in a directed acyclic graph.

Section II describes related work in the context of ontologies and IT management, and section III gives an overview of our approach. The paper closes with a conclusion in section IV.

II. Related Work

There are several publications that examine the application of ontologies to the domain of IT management, e.g. [4], [5]. In [6] the authors provide mappings for parts of different IT management models to OWL, including Structure of Management Information (SMI) and the Common Information Model (CIM). The resulting ontology can be used to combine the knowledge given in the different representations into a joint model. One problem the authors point out for the mapping is information that can be expressed in the original languages, but has no direct representation in OWL, such as the attachment of measurement units or access authorizations

 ISBN: 978-1-61208-175-5

to properties. To solve this problem, the data is presented on the Resource Description Framework (RDF) layer of OWL. In RDF, it is possible to attach additional information to edges in the graph so that the data can be represented.

[4] describes how to represent several abstraction layers of a system in split ontologies to achieve a pyramid-like structure of ontologies, where often used ontologies are at the bottom of the figure. The re-use of components and models is always an important topic in IT systems. The paper shows that OWL is capable of organizing several abstractions of a system in ontologies and reuse defined components in higher layers.

A real-world management application is shown in [5] where ontologies are used to manage a network infrastructure. SWRL rules are used to create new object property connections between entities in case of a blackout. For this, properties and instance structures are observed. As basis for the paper Policy-based Network Management (PBNM) [7] was used. Rules are evaluated periodically during runtime, and new facts are added to the ontology. A management component observes the ontology and maps newly added facts to management operations to adjust the system.

There are no other methods known to the authors for the combination of ontologies and Bayesian networks in an IT management context, but there are approaches to embed probabilistics into OWL. In [8] the embedding of probabilistic knowledge for OWL class membership is presented. The major problems are the representation of probabilistic knowledge in OWL, the derivation of an acyclic graph and the construction of conditional probability tables. Therefore, special OWL classes are defined to represent the expressions $P(A)$, $P(A|B)$ and $P(A|\overline{B})$, which have properties for conditions, values and probabilities. These properties are used to generate the conditional probability tables. A specially modified reasoner is needed to evaluate the ontology, as the existing reasoners cannot be used.

One problem that has to be taken into account when updating facts in a knowledge base, is that the knowledge base may enter an inconsistent state because of previously derived facts contradicting the changes. This is known as *belief change*, and in the context of ontologies, as *ontology change*. Several works approach this problem, e.g. [9], where the authors examine the applicability of solutions from belief change theory to ontologies. Another approach to the problem is taken in [10], which proposes an ontology update framework where ontology update specifications describe certain change patterns that can be performed.

III. Architecture

A new architecture for ontology-based automated IT management is currently under development by the authors and the main ideas are sketched in this section. The architecture consists of a set of components (shown in Figure 1), which can be grouped into

- Importers that add new data to the ontology
- Reasoning components, which use the existing data to derive new knowledge
- Management components, which interact with the system under management.

The central element of the system is an ontology that is used as a shared knowledge base (blackboard) for all components. Each component can read data from the knowledge base and add or remove facts from it. Services are used for the inter-component communication. The architecture is designed to be used in a distributed fashion.

A. Importers

The combination of different domain models raises the requirement for corresponding importers. These specific components know how to map the domain specific model to an ontology model. Hence, an interface is defined, which allows the use of new domain specific model importers. Implemented model importers are an ontology importer and a CIM importer. The ontology importer simply reads the data from an OWL ontology and adds the facts to the shared knowledge base. The CIM importer uses the mapping rules described in [3] to map the CIM schema to OWL facts.

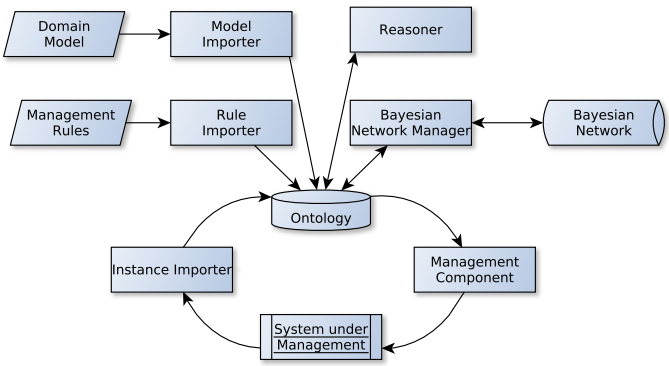


Figure 1. Components of the developed architecture

As well as models, rules can be specified in a domain specific manner. Hence, an interface is provided for the implementation of domain specific rule importers. Internally, SWRL is used as rule format for the shared ontology and an according importer was implemented.

In general, the domain model contains just the taxonomy of the monitored system but not the instance data. Therefore, a component is needed that monitors the system under management and imports runtime data into the ontology by creating according instances. Such components are called instance importers. An interface is provided for the integration of domain specific instance importers. Already implemented instance importers are the log record importer, which maps log records to instances and relations, and the CIM instance importer, which uses the OpenPegasus CIMOM to get information from a CIM-based management

 ISBN: 978-1-61208-175-5

system. Other application-specific instance importers can be added as needed.

B. Reasoning

The strength of OWL and its formal grounding is the ability to reason new knowledge from an existing knowledge base. In our architecture this feature is used to derive new facts from the domain specific models, the imported rules and the monitored instance data.

In many cases it is insufficient to just consider exact knowledge in IT management, because side effects and complex relationships are either not known or can not be modeled in an according abstraction. But especially for state prediction and root cause analysis probabilistic knowledge and the statistical consideration of historical data is needed. Because of that, a concept is used to make probabilistic modeling and reasoning possible, which is described in detail in [11]. The structure of the Bayesian network is derived from the OWL model. Specially annotated OWL instances become nodes and specially annoted OWL properties become arcs in the Bayesian model. The joint distribution tables are not modeled in the ontology directly, but trained using a maximum likelihood algorithm during a precedent training phase.

Ontologies are able to represent continuous and discrete variables, in OWL this is done using data properties. As Bayesian networks only work on discrete random variables, a discretization must be applied. To discretize continuous variables, some additional information is needed. OWL does not support the addition of supplemental data to data property assertions. Hence, a special variable class is defined, which has a data property that contains the actual value of the variable. There are three different types of variables: Continuous variables, Discrete variables and Enumerations. A mechanism is needed to map values of all three types of variables from the ontology to the generated Bayesian network and back again. Since enumerations generally have just a small state space, the values can be mapped one by one. For continuous and discrete variables the mapping is problematic and a discretization must be applied.

Because causal relationships can be seen as unidirectional edges between entities, the OWL object property concept can be used for their representation. In general it is not possible to connect data properties in OWL, but in this case it is feasible because all variables are already encapsulated by instances of the variable class.

For the evaluation of these relationships, causations are mapped to a Bayesian network where each instance of the variable class becomes a node. For numerical variables each variable is checked for intervals. A discrete state is created for each interval in the state space of the node in the network. Enumerations are checked for their defined enumeration class and for each individual of this class a state is created with the unique name of the individual. Causal relationships between variables become arcs in the Bayesian network.

In the next step the OWL model is analyzed for variable states, which will be set as evidences in the Bayesian network. Subsequently, an inference algorithm is applied to calculate the belief for the states of unobserved variables (variables which have no value set in the ontology). If the calculated belief is above a defined threshold, the deduced value is set for the variable in the ontology and can thereby be used by the exact reasoners for further reasoning. To ensure the knowledge exchange between the reasoning components a component can be called multiple times in a reasoning cycle.

C. Management components

Management components are used to reconfigure the system under management. They contain the knowledge that is needed to interact with a specific component of the system. Depending on the evaluation results of the rules, according actions are triggered. When CIM is used as a domain model, the management components can call methods on the CIMOM, which in turn controls the particular component, or it can execute external commands directly.

D. Runtime

The first step on application startup is the import of required domain models and rules using the according model and rule importers. After that, the management cycle is started (also known as MAPE-K loop [12], which stands for monitor, analyze, plan, execute and knowledge). The loop begins with the monitoring phase, where information from the system under management is read and imported into the ontology as instances.

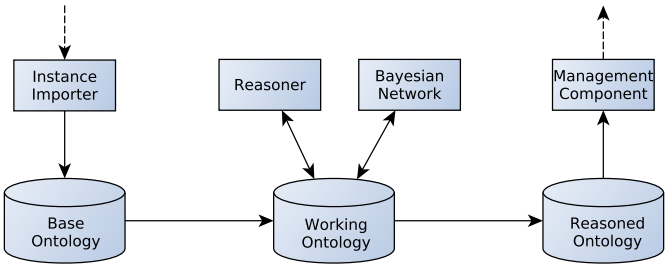


Figure 2. Multi step ontology reasoning process

In the analysis phase, the domain models, the rules and the monitored data are used for the reasoning of new knowledge. The reasoning process is shown in Figure 2.

The base ontology contains all the imported and monitored data. When the reasoning process starts, all data of the base ontology is copied into the working ontology. All reasoners are applied to this ontology sequentially and add their reasoned knowledge to it. When all reasoners have finished, the data of the working ontology is copied to

 ISBN: 978-1-61208-175-5

the reasoned ontology, which is used for queries into the knowledge base and stays untouched until the next reasoning phase has finished.

The reasoning takes place in this multi-step process for two reasons: The first reason is handling ontology change, as new information can be added easily to an ontology, but not retracted easily. By keeping the base model and inferred knowledge from different reasoners in separate sub-ontologies, inferred knowledge from a single reasoner can be retracted without effort. The second reason is that the last version of the *reasoned ontology* can still be queried, while the new version is being created. As reasoning can be slow on large ontologies, this makes sure that clients do not block on queries but can always receive an instant reply. The query result therefore may be as old as one reasoning cycle.

The last steps in the cycle are the plan and execute phases. The management components use the data of the reasoned ontology to make management decisions and execute them on the system under management. The presented architecture is partially implemented in Java using the OSGi Framework as service middleware. For the service abstraction the interfaces of the OWL API are used.

IV. Conclusion

In this paper we sketched an approach for ontology-based IT management. An architecture that uses an ontology combined of the domain model, rules and dynamically updated instance data was presented. Two main problems must be solved: The first problem is the creation of a suitable domain model, which was covered by the translation of CIM to OWL and the expression of probabilistic knowledge using Bayesian networks. The integration of other domain models has yet to be examined. The second problem is the continuous update of the ontology with new facts. This is a topic of current research, and our solution is a multi-step reasoning process. Performance comparisons to other approaches and with different ontologies must be conducted.

Future work includes the development of importers for other domain models. It also includes the application of the developed tool on storage management and the ambient assisted living (AAL) context. Furthermore, performance needs to be optimized.

In the context of storage management the Storage Management Initiative Specification (SMI-S), which is a specialization of the CIM Model, can be used to manage storage systems. Rules, which are verbally defined in the specification, can be formalized and integrated into the OWL model. Besides, the probabilistic part can be used to make assertions about future states (e.g. how high is the probability of a full file system tomorrow if there is a peak) or to analyze previous scenarios (e.g. what was the most likely reason for a file server crash). In combination a pro-active management can be achieved and systems can be reconfigured before an error occurs.

In the context of ambient assisted living the domain will be a living environment, equipped with a set of sensors and effectors. That environment will be modeled in a hierarchy of ontologies and monitored during runtime. The observed data is used to derive higher level knowledge, e.g. that an elderly person lies on the ground and needs help.

References

[1] Distributed Management Task Force, “Common Information Model (CIM),” http://www.dmtf.org/standards/cim/.

[2] World Wide Web Consortium, “OWL Web Ontology Language,” http://www.w3.org/TR/owl2-overview/.

[3] A. Textor, J. Stynes, and R. Kroeger, “Transformation of the Common Information Model to OWL,” in *10th International Conference on Web Engineering - ICWE 2010 Workshops*, ser. LNCS, vol. 6385. Springer Verlag, July 2010, pp. 163–174.

[4] J. E. L. De Vergara, A. Guerrero, V. A. Villagrá, and J. Berrocal, “Ontology-Based Network Management: Study Cases and Lessons Learned,” *Journal of Network and Systems Management*, vol. 17, no. 3, pp. 234–254, 2009.

[5] A. Guerrero, V. A. Villagrá, J. E. L. de Vergara, A. Sánchez-Macián, and J. Berrocal, “Ontology-Based Policy Refinement Using SWRL Rules for Management Information Definitions in OWL,” *Large Scale Management of Distributed Systems*, vol. 4269, pp. 227–232, 2006.

[6] J. E. L. De Vergara, V. A. Villagrá, and J. Berrocal, “Applying the Web ontology language to management information definitions,” *IEEE Communications Magazine*, vol. 42, no. 7, pp. 68–74, July 2004.

[7] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser, “Terminology for policy-based management,” United States, 2001.

[8] Z. Ding and Y. Peng, “A probabilistic extension to ontology language owl,” in *In Proceedings of the 37th Hawaii International Conference On System Sciences (HICSS-37), Big Island*, 2004.

[9] G. Qi and F. Yang, “A survey of revision approaches in description logics,” in *Web Reasoning and Rule Systems*, ser. LNCS, vol. 5341. Springer, 2008, pp. 74–88.

[10] U. Lösch, S. Rudolph, D. Vrandečić, and R. Studer, “Tempus fugit - towards an ontology update language,” in *6th European Semantic Web Conference (ESWC 09)*, vol. 1. Springer, January 2009, pp. 278–292.

[11] F. Meyer, “Kombination von Modellen zur Systemanalyse im Selbstmanagement-Kontext,” in *Workshop Self-Organising, Adaptive, Context-Sensitive Distributed Systems (SAKS 2011)*, ser. Electronic Communications of the EASST, March 2011.

[12] IBM Corporation, “An Architectural Blueprint for Autonomic Computing, Technical Whitepaper (Fourth Edition),” June 2006.

 ISBN: 978-1-61208-175-5

A Multi-Layer Approach to the Derivation of Schema Components of Ontologies from German Text

Mihaela Vela and Thierry Declerck
Language Technology Lab
DFKI Saarbrücken
Saarbrücken, Germany
Mihaela.Vela@dfki.de Thierry.Declerck@dfki.de

***Abstract*—We describe an on-going work on the semi-automatic derivation of ontological structures from text. Hereby, we first apply on plain text pattern-based linguistic heuristics, for identifying relevant segments out of which candidate ontology classes and relations can be derived. The second step proposes a consolidation of those candidates on the basis of a partial linguistic and semantic analysis of the textual context of the segments. The last step is dealing with the extension of the derived ontology structures. We use for this a constituency and dependency analysis of the textual segments selected in steps 1 and 2. We show how these three steps support in different but related ways the derivation of ontology components from text.**

Keywords – knowledge acquisition; text-based knowledge

I. INTRODUCTION

We describe a semi-automatic incremental multi-layer rule-based methodology for the derivation of ontology schema components from a corpus consisting of the 1992 edition of the German newspaper "Wirtschaftswoche". We use this somehow older corpus, since it has been manually annotated with various types of information. The corpus comprises 200107 words, 11583 sentences and 121331 phrases. By *Derivation of Ontology Schema Components* we mean the acquisition from text of possible concepts and relations between these concepts for the semi-automatic ontology building. By *Ontology Schema* we mean a construct similar to the T-Box of an ontology [23]. Our work is addressing the intensional part of ontologies and can be considered as contributing to the ontology learning field at large. Ontology learning is the process of semi-automatic support in ontology development (see [1]).

We are dealing in our work primarily with German text. In this concrete case, we consider compound nouns and their paraphrases in the corpus as the basic segments in text that can serve for the detection of candidate ontology classes and relations. Compounding is a very rich word formation process in German (and other related Germanic languages), also with well-established construction patterns corresponding to semantic types, which makes them good candidates for the derivation of ontology schema components. We use paraphrases of nominal compounds in the corpus for fixing their status as candidates for classes and for specifying the relations existing between those classes. Paraphrases of compounds are defined as a text segment containing the elements of the compound nouns separated by a limited number of other word forms.

In a second step, we apply morphological, Part-of-Speech (PoS) and lexical-semantic analysis to the text segments described in step1. This helps further filtering out and further specifying the previously derived candidates, avoiding redundancies in the derivation of classes (limiting the names of class labels to lemmas, and joining labels that are synonyms, etc.)

In the last step, we extend the extracted classes and relations on the basis of deeper linguistic processing, more precisely analyzing the constituency and dependency structures of the context of the detected textual segments. Our approach results in a set of generic patterns (in machine learning language we would call them seeds) for deriving a stable structure of conceptual relations from the combined shallow and linguistic analysis of specific textual segments.

The paper is structured as follows. Section 2 gives an overview on related work. Section 3 describes the pattern-based processing of text for detecting segments containing candidates for ontology derivation. Section 4 presents the ontology derivation potential from the textual context of the segments, annotated with PoS, morphology, and lexical semantics. Section 5 deals with the refinement of the ontology derived so far, using constituency and dependency information. Section 6 describes some evaluation work and Section 7 concludes and names some issues for further work.

II. RELATED WORK

There are purely linguistic approaches to Ontology Learning ([3][4][5]), linguistic approaches making use of machine learning for generalization ([6]) and machine learning approaches that use linguistic information ([2][7]). Those approaches have in common that they concentrate on discovering new relations, although some approaches are dealing with the discovery of new concepts ([2][6][8]) too.

The purely linguistic approaches ([3][4][5]) perform ontology learning on the basis of deep linguistic analysis, by activating a graphical interface controlled by the user for entering the extracted knowledge into the ontology.

The method proposed in this paper is based on linguistic patterns, combining shallow and deep linguistic analysis, in an unsupervised way, and thus not involving authoring tools.

 ISBN: 978-1-61208-175-5

Our work resembles most the one presented by [6], but our combination from shallow and deep linguistic analysis allows covering a wider range of phenomena for the derivation of schema components of ontologies.

III. PATTERN-BASED TEXT ANALYSIS

Although pattern-based linguistic heuristics alone is not enough to acquire extended and complex ontological knowledge, a pre-processing of the plain text is very important when it comes to define an anchor (text segment) from which to start the computationally more expensive process of ontology learning.

A. Detection of Candidate Concepts and Relations

A first intuition guiding our investigation is the fact that German nominal compounds are good indicators for the expression of relations between concepts expressed by the elements of the compounds. According to [9], the German determinative compounds (determinative compounds are those in which one element is subordinated to the other element of the other, more precisely, one element determines/specifies the other element [10]) consist mostly of two elements, whereas the first one usually specifies the second. From this observation one can heuristically derive a hyponymy relation between the whole compound and its second element: *Konzernchef* (*chief of corporation*) is a specific type of a C*hef (chief)*.

Although German uses also copulative compounds, we do not expand on those in the actual paper, in which we concentrate on binary determinative noun-noun compounds (copulative compounds are compounds were the elements are considered semantically coequal and which do not have a main element which specifies or determines the other element in the compound. This type of compounding is more seldom in German [11]). We implemented a quite straightforward pattern-based algorithm for the detection of this type of compounds: we first search for nouns in the corpus (for German, a string starting with a capital letter between blanks or between a blank and a punctuation sign). If, in a second search round, we can detect that such a noun item appears as sub-string in a larger noun, then we considered that we have found a compound. While this approach works quite well for finding the nouns acting as the prefix of a compound (since it starts with a capital letter), we need to access a lexicon for deciding if the suffix of the compound is also a noun (we use for this the lexicon listed in [24].)

We include in our patterns the German joint elements (Fugeelement) which may appear in compounds (such as "*s*" in *Wohnungsbau* (*house building*), in order to get the right string, when the word is used in isolation. But with our very simple approach we do miss the nouns that undergo morphology changes when they are used in a compound.

We consider the two elements of a nominal compound as acting as potential ontology classes, and the remaining task is then to specify the possible relations between these two nouns, or candidate ontology classes.

B. Deriving Candidate Ontology Classes and Relations from Nominal Compounds

On the basis of the detection of compounds, and assuming that elements of compounds act as possible ontology classes, we suggest two rules for deriving potential elements for the schema of an ontology: the structural type represented by the *subClassOf* relation (rendering the relation between the whole compound and its second element) and a relation denoting an *objectProperty* (rendering the relation between the two elements of the compound). We are using here the OWL-DL terminology for the property name.

The first rule states that between a compound as a whole and its second noun there is a subClassOf relation. This decision is motivated by the definition of the determinative compounds which introduces hyponymy between the compound and its second noun.

For example, from the compound *Bankenvertreter* we derive the relation: *subClassOf(Bankenvertreter, Vertreter)*, which translated into English means that a *representative of a bank* is a *subClassOf* a *representative*.

Our intuition - sustained by the already existing analyses of the German compound ([11][12][13]) - was that there exists also an additional relationship between the elements of a compound, which we consider of being of type *objectProperty*. Applying the corresponding rule to the already mentioned compound *Bankvertreter* we derive a *objectProperty(Bank, Vertreter)* relation between the class *Bank (bank)* and the class *Vetreter (representative).*

Obviously, the (naïve) processing strategy presented above is very general and the very generic *objectProperty* relation we can derive is not really satisfying. In order to improve this state, we try to find expressions in the text that are containing paraphrases of the compounds, expecting to find more semantic information for allowing the further specification of the (generic) object property relation we established between the elements of a compound.

C. Patterns for the Recognition of Paraphrases of Compounds

After splitting the compound back into *noun1 + noun2* we automatically search for paraphrases (in our context a paraphrase consists of a test window that contains the elements of a compound separately) of all found compounds in our corpus. Our decision to look for the paraphrases of compounds is motivated by the fact, that while we assume that the elements of a compound are semantically related to each other, analyzing the paraphrases will allow specifying more precisely this relation [9]. Compounds without a paraphrase are no longer considered for ontology derivation. For now the search space for detecting paraphrases is our corpus, but this will be extended to other corpora.

Our assumption is also sustained by [11] and [13]. Although they have two different methods for approaching this issue, the main idea is the same: the elements of a compound are semantically related to each other and this relation becomes visible in the paraphrase.

We find in the corpus two kinds of paraphrases, in which the elements of the original compounds are linguistically

 ISBN: 978-1-61208-175-5

related: either by a genitive article as *Vertreter der Bank (representative of the bank)* or by a preposition as *Chef im Konzern (chief of corporation).* The finding of a paraphrase for a compound validates the *subClassOf* relation, whereas the use of lexical semantics on the elements of a paraphrase allows specifying the *objectProperty*.

IV. SHALLOW LINGUISTIC ANALYSIS

The addition of PoS and morphology annotation to the paraphrases helps in solving the redundancy problem of the ontology classes: by using lemmas for generating names of classes we avoid generating as many classes as this lemma has morphological variations in the text. Lexical semantics allows reducing the number of classes by grouping lemmas to more general "words" (like the synsets of GermaNet (GN) [14]) and at the same time specifying the derived generic relation *objectProperty* according to the semantics (therefore we use GN's semantic fields for nouns: artifact, attribute, shape, feeling, body, cognition, communication, motive, food, object, phenomenon, plant, substance, time, animal, state, act, process, person, group, possession, relation, attribute, event, quantity, location) of these lemmas and of other word forms present in the paraphrase.

A. Specifying Relations with Lexical Semantics

Analyzing the paraphrases annotated with GN's semantic information we discovered the following six relations between the already detected classes:

- *hasPosition,*
- *disposesOver,*
- *hasDimension,*
- *hasAttribute,*
- *hasEvent,*
- *hasLocation.*

For example, for the compound *Aktiengesellschaft (stock company)* we found the reformulation *Aktien der Gesellschaft (shares of the company),* where *Aktien* was semantically classified as belonging to GN's semantic class *possession* and *Gesellschaft* has been classified as belonging to GN's semantic class *group* enabling the structural integration of the discovered classes and relations into a more sophisticated ontology structure. The heuristics for the derivation of the relation between the two concepts *Aktien (shares*) and *Gesellschaft (company)* proposes the verbalization of the more generic class to which the first noun in the paraphrase belongs. This way the verbalized *possession* was transformed into *disposesOver* generating *disposesOver(Gesellschaft, Aktien).*

Applying morphology and lexical semantics to the second type of paraphrase patterns, those involving prepositions, we notice that the generic *objectProperty* can be further specialized depending on the lexical semantics of the used prepositions.

Prepositions are semantically ambiguous, but the ambiguity can be reduced on the base of the lexical semantics of the associated nouns. Analyzing this type of paraphrases we discovered, based on the same heuristics as for genitive phrases, a set of six rules for the derivation of ontological relations. From this six relations, five were already discovered during the analysis of genitive phrases: *disposesOver, hasDimension, hasAttribute, hasEvent,* and *hasLocation.* Only one relation is new: the *hasAffiliation* relation.

B. Analyzing Modification Phenomena

In the process of detecting paraphrases we observed that many of the paraphrases contain modifiers. In order to determine the type of ontological relation that can be extracted from the structure modifier(s)-nominal head (such as *multinationale Gesellschaft* (*multinational corporation*)), some components of the structure had to be viewed from a lexical semantic point of view. We concentrate here on adjectives and adverbs, and apply to them various language specific classification schemes.

For adjectives we used the classification by [15] and for adverbs the classification by [16] (we use for modifiers this semantic classification because they are more fine-grained than GermaNet's classification and we can easily add new adjectives and adverbs to it). As for nouns, the semantic classes to which the adjectives and adverbs belong are introduced as ontology classes.

Based on this classification we introduce new relations between the modifiers and the noun they modify. Having for example the paraphrase *Aktien der deutschen Gesellschaft* (*shares of the German corporation*), the derivation rule will return the following relation: *hasNationality(Gesellschaft, Nationality).* Here *hasNationality* is a subproperty of *hasAffiliation.*

Many of the nouns appearing in paraphrases are modified by just one modifier. But there are cases in the corpus in which a noun is preceded by more than one modifier. For multiple premodifiers which are not separated by any punctuation sign or conjunction to each other, we speak of an aggregation of adjectives. For example for *großen deutschen Konzern* (*large German concern*), linguistically the first premodifier in the token chain modifies the remaining phrase [17]. From this kind of linguistic constructions we extract *hasNationality(Konzern, Nationality)* and *hasDimension(Konzern, Dimension).*

A different linguistic principle applies for modifiers connected by punctuation signs or/and conjunctions: each pre-modifier introduces a relation between itself and the noun it modifies [17]. From *kleinen, krisengeplagten Firmen* (*small firms, affected by the crisis*) we extract *hasDimension(Firma, Dimension)* and *hasMode(Firma, Mode).* As one can see, we cannot model directly the two different ways plural modification is linguistically working in the ontology.

A more specific case is represented by the modification of adjectives by adverbs such in *sehr großes Gehalt* (*very big salary*). In this case the adverb *sehr* modifies the adjective *großes* and not the whole phrase [10] *großes Gehalt.* We extract then the relations: *hasAspect(Dimension, Aspect)* and *hasDimension(Gehalt, Dimension).*

Since modification is a very powerful linguistic phenomenon with a high coverage in the corpus, the three

 ISBN: 978-1-61208-175-5

extraction rules discussed above cover 26 relations, depending on the semantic class of the modifier.

V. PHRASE STRUCTURE AND SYNTACTIC INFORMATION

Although, many extraction rules were generated with the shallow linguistic analysis, we are aware that the sentential level is an additional resource for the extraction of ontological information. We decided to first analyze predicate-argument structures in all sentences containing a paraphrase, since those contain in our sense already enough hints for possible ontology classes and relations. The analysis of the extracted sentences has shown that there is potential for extracting additional ontology schema components. In this case we also have to take into account additional PoS tags and morphological information (for example for the verbs). As a lexical-semantic resource for the verb, we use both the classification by [18] and GN.

A. Extraction of Ontology Schema Components from Grammatical Functions

With the help of grammatical functions (for example the subject-object relation in a sentence) we developed a set of rules for extracting the arguments of specific verbs in the corpus. This allows extracting relations such as

- *earn*,
- *appliesFor*,
- *estimate*,
- *hasPossession*,
- *partOf*,
- *subClassOf*,
- *etc.*

Let us consider the following sentence: *Die Papierherstellung ist zu einer extrem kapitalintensiven Branche geworden* (*Paper production evolved to a very capital-intensive branch*). In this example, the verb *sein* (*be*) connects the subject *Papierherstellung* (*paper production*) and the *kapitalintensiven Branche* (*capital-intensive branch*) of the sentence.

In fact, the rule states that only the nominal heads of the phrases identified as subject and object enter the ontology and therefore we extract *subClassOf(Papierherstellung, Branche*). Additionally, for each of the two nouns we use GN's information about synonyms, antonyms, hyponyms and meronyms. In a next step, we include then also the information that *Branche* can have the property *kapitalintensiv*.

VI. EVALUATION

The evaluation of the method for extracting ontology schema components was performed on a manually annotated test suite. The test suite consists of 200 randomly selected sentences (out of over 11000) which were annotated by a student of business informatics. We plan to ask another person to annotate the same corpus. This was till now not possible for time reasons.

We applied our method and the corresponding tools on this test corpus. The quantitative evaluation was performed in two stages, and after each stage we measured the performance of our method. We compared the results of our method with the manual annotation by calculating the F-measure.

TABLE 1. PRECISION AND RECALL SCORES

Phenomenon	Prec.	Recall	1st run	2nd run
Compound (*subClassOf*)	1	1	1	1
Modification	1	0,52	0,68	1
(Para)phrase	1	0,23	0,37	0,76
Gramm Funct.	0.5	0,30	0,38	0,80

From the results in Table 1 we notice that we have the best results when it comes to extract the *subClassOf* relation, which is extracted mainly from compounds. It seems that our compound filtering process is really helping in getting a high number of correct answers. But it seems also that the 200 manually annotated sentences contain only determinative compounds, and we would have to test our method on copulative compounds too.

The *subClassOf* relation is extracted not only from compounds but is introduced into the ontology from GN (using the more abstract "words" in the synsets). In this case the left-hand side argument of the *subClassOf* relation differs from the one chosen by the manual annotator.

We consider still our method to be valid, since we found it totally normal that a human being annotates semantically different than GN (the student didn't have GN as a resource to consult for his annotation). Both assignments by GN and by the student are correct, but we notice that the manual annotator has chosen a more specific class than the one our method uses.

The results from the modification phenomena show that we have a very good precision. This means that we either find a true relation or we do not find it at all. This corresponds to the methodology applied: if a modifier is in our modifier lexicon it produces a true relation, if not it does not produce anything and these we can read from the recall score.

For the relations extracted from phrases we achieve the lowest scores concerning the recall. This low score is due to three factors: there is no rule for extracting a relation, the implemented rule does not work properly and the rule exists but it does not fire because of lack of semantic information. We can influence on the first two factors by writing new rules or improving the implementation of the existing rules.

In fact the GN lookup fails because certain nouns in our analysis do not have a stem and the GN lookup is based on stems. This is an issue that we can solve in a next stage of our work.

The scores for ontology extraction from grammatical functions show one characteristic common to all other phenomena: the relation is either not found but if it is found than it is correct. The precision and recall (and consequently the F-measure) scores are influenced not necessary by our

 ISBN: 978-1-61208-175-5

rules, but by the assignment of grammatical functions by the parser. Because we cannot influence the ambiguity of the grammatical function assignment, in the second evaluation round we manually corrected the ambiguities provided by the parser.

In the second evaluation round we concentrated also on relations from phrases and modification. We improved the scripts implementing the rules for ontology extraction from phrases and enlarged our lexicons for ontology extraction from modification phenomena. We also have to notice here, that the disambiguation of the grammatical function assignment provided a considerably improvement of the measured scores.

Also part of the evaluation, in a broader sense, is the integration of the ontological knowledge extracted here into a bigger ontology. We suggest for this purpose The Suggested Upper Merged Ontology (SUMO) [25]. SUMO is in a large freely available ontology. Another important characteristic of SUMO is the fact that it has been mapped to the whole lexicon of WordNet. From this perspective, SUMO is the ontology which fits our approach, when it comes to integrate our work into a broader ontology. It is true, that there is no direct mapping between GN and SUMO. This situation can besolved by first mapping from GN to WordNet and then to SUMO. The direct mapping between GN and WordNet is possible since both have the same general structure concerning the semantic tree.

VII. CONCLUSION AND FURTHER WORK

Our aim was to present a multi-layer, rule-based approach for the extraction of ontology schema components and to show that a significant amount of ontological knowledge can be derived without using exclusively deeper linguistic information.

While applying our method on German language, we saw that this approach can be extended to all Germanic languages making use of compounding. Swedish is a good example, and [22], for example discusses the potential of compound for building a FrameNet resource for Swedish.

We also experimented with other language families, more specifically French. Different from the German compounds, the French compounds are not always conflated to a single word. The cumulated form of compounds such as *sociolinguistique (sociolinguistic)* is in French the exception. The majority of compounds in French consist either of two components connected by a hyphen such as *timbre-poste (stamp)* or are just two or more words which appear in a lexical chain such as *dessin animé (animated cartoon)* or *séance marathon (marathon session)*. The most productive of the latter compounds are the compounds constructed with prepositions such as *mesure de sécurité (safety measure)*. Noun-noun compounds are in French less frequent than in German or English [21]. We applied our method on compounds consisting of nouns, of an adjective and a noun and prepositional compounds. Assuming the appropriate linguistic tools, our method can be applied to French text.

It seems thus that only the first step of our work would need a complete re-implementation when applying our strategy to other language (families).

The phenomena which we consider in this work are compounding, nominalization, premodification, postmodification, phrase-structure combined with lexical semantics. From the purely linguistic point of view we do not take into consideration the peculiarities of relative clauses. We also do not cope now with the semantic and linguistic properties of the negation particle or with coreference. These phenomena are not treated here because of a more pragmatical and practical reason: the linguistic tools we have at hand do not annotate these kinds of phenomena. Experiments on the instantiation were also performed, achieving promising results. To integrate these phenomena into the approach presented here remains an issue for future work.

Beside these points, we are now working on modeling our findings about the relations between natural language expressions and ontology schema components in an appropriate way. This is done within the context of a running European R&D project, the Monnet project [26]. In this project, a model, called “lemon” [27], for representing lexicons in ontologies, has been implemented. While this model has been primarily designed for the ontological representation of natural language expressions used in labels of ontologies, we see a big opportunity for using this model for the representation of language data we have been dealing with in the context of knowledge acquisition from text. First steps are dealing with abstracting over the lexical material we found in text, and confining ourselves with the use of linguistic categories, that are related to specific ontology schema components. The work is thus going toward a declarative description of linguistic patterns that should be used in ontology engineering.

VIII. ACKNOWLEDGEMENTS

The work presented in this paper was supported (in part) by the European project MONNET No. (FP7/2007-2013) 248458.

REFERENCES

[1] P. Buitelaar, P. Cimiano, and B. Magnini, “Ontology learning from text: An overview,” in Ontology Learning from Text: Methods, Evaluation and Applications. Frontiers in Artificial Intelligence and Applications, IOS Press, vol. 123, pp. 3-12, 2005.

[2] M. A. Hearst, “Automatic acquisition of hyponyms from large text corpora,” in Proceedings of 14th International Conference on Computational Linguistics (COLING-92), pp. 539-545, Nantes, France, 2002.

[3] P. Cimiano, A. Hotho, and S. Staab, “Learning concept hierarchies from text corpora using formal concept analysis,” Journal of Artificial Intelligence Research, vol. 24, pp. 462-471, 2005.

[4] G. Aguado de Cea, A. Gomez-Perez, E. Montiel-Ponsoda, and M. del Carmen Su´arez-Figueroa, “Natural language-based approach for helping in the reuse of ontology design patterns,” Proceedings of the EKAW Conference, pp. 32-47, 2008.

[5] N. Aussenac-Gilles and M.-P. Jacques, “Designing and evaluating patterns for relation acquisition from texts with caméléon,” Terminology, vol. 5, nr. 1, pp. 450-473, 2008.

 ISBN: 978-1-61208-175-5

[6] M. Ciaramita, A. Gangemi, E. Ratsch, J. Saric, and I. Rojas, "Unsupervised learning of semantic relations for molecular biology ontologies," in Ontology Learning and Population: Bridging the Gap between Text and Knowledge, IOS Press, pp. 91-107, 2008.

[7] P. Gamallo, M. Gonzalez, A. Agustini, G. Lopes, and V. S. de Lime, "Mapping syntactic dependencies onto semantic relations," in Proceedings of the ECAI Workshop on Machine Learning and Natural Language Processing for Ontology Engineering, pp. 15-22, 2002.

[8] E.de la Clergerie, "Spécifications du service d'extraction supervisée d'ontologies," SCRIBO Project, Tech. Rep., 2009.

[9] J. Erben, "Einführung in die deutsche Wortbildungslehre", 3rd ed. Berlin: Erich Schmidt Verlag GmbH and Co., 1993.

[10] Duden, "Grammatik der deutschen Gegenwartssprache", 4th ed Mannheim/Wien/Zürich, Dudenverlag, 2006.

[11] M. Lohde, "Wortbildung des modernen Deutschen", Tübingen: Francke, 2006.

[12] W. Fleischer and I. Barz, "Wortbildung der deutschen Gegenwartssprache", 2nd ed. Tuebingen, Niemeyer, 1995.

[13] W. Motsch, "Deutsche Wortbildung in Grundzügen", 2nd ed. Berlin, de Gruyter, 2006.

[14] C. Kunze and L. Lemnitzer, "GermaNet – representation, visualization, application," in Proceedings of the LREC, pp.1485-1491, 2002.

[15] S.-M. Lee, "Untersuchungen zur Valenz des Adjektivs in der deutschen Gegenwartssprache", Frankfurt am Main, Germany: Lang, 1994.

[16] A. Lobeck, "Discovering Grammar: An Introduction to English Sentence Structure", Oxford University Press, 2000.

[17] G. Zifonun, L. Hoffmann, and B. Strecke, "Grammatik der deutschen Sprache", Berlin/New York, de Gruyter, 1997, vol. 3.

[18] H. Schumacher, Verben in Feldern, 1st ed. Berlin: de Gruyter, 1986.

[19] M. Vela and T. Declerck,"A Methodology for Ontology Learning: Deriving Ontology Schema Components from Unstructured Text", Proceedings of the Workshop on Semantic Authoring, Annotation and Knowledge Markup, Redondo Beach, California, United States, pp. 22-26,2009.

[20] M. Vela and T. Declerck, "Concept and Relation Extraction in the Finance Domain", Proceedings of the Eighth International Conference on Computational Semantics (IWCS-8), Tilburg, Netherlands, Tilburg University, pp. 346-351, 1/2009.

[21] H. Geckelerand and W. Dietrich, "Einführung in die französische Sprachwissenschaft", Erich Schmidt Verlag, 2007.

[22] L. Borin, D. Dannélls, M. Forsberg, M. Toporowska Gronostaj, and D. Kokkinakis, "The Past Meets the Present in the Swedish FrameNet++" , 14th EURALEX International Congress, pp. 269-281, 2010.

[23] http://wiki.dbpedia.org/Ontology, retrieved October, 2011.

[24] http://wortschatz.uni-leipzig.de, retrieved October, 2011.

[25] http://www.ontologyportal.org/, retrieved October, 2011.

[26] www.monnet-project.eu/, retrieved October, 2011.

[27] J. McCrae, G. Aguado-de-Cea, P. Buitelaar, P. Cimiano, T. Declerck, A. Gomez-Perez, J. Gracia, L. Hollink, E. Montiel-Ponsoda, and D. Spohr, "Interchanging lexical resources on the Semantic Web. Journal for Language Resources and Evaluation", in press.

 ISBN: 978-1-61208-175-5

A Linked Dataverse Knows Better: Boosting Recommendation Quality Using Semantic Knowledge

Andreas Lommatzsch, Till Plumbaum, and Sahin Albayrak
Technische Universität Berlin, DAI-Labor
Ernst-Reuter-Platz 7, D-10587 Berlin, Germany
{*andreas.lommatzsch, till.plumbaum, sahin.albayrak*}*@dai-labor.de*

***Abstract*—The advent of Linked Open Data (LOD) gave birth to a plethora of open datasets freely available to everyone. Accompanied with LOD, a new research field arises focusing on how to handle and to take advantage of this huge amount of data. In this paper, we introduce a novel approach utilizing and aggregating open datasets to compute the most-related entities for a set of weighted input entities. We optimize different algorithms for large semantic datasets enabling combining data from different semantic open sources and providing high quality results even if only limited resources are available. We evaluate our approach on a large encyclopedic dataset. The evaluation results show that our approach efficiently supports different semantic edge types. The application build on our framework provides highly relevant results and visual explanations helping the user to understand the semantic relationship between the computed entities.**

***Keywords*-linked open data; recommendation; semantic web; user profile enrichment; personalization**

I. INTRODUCTION

With the rapidly growing number of large open datasets following the Linked Open Data (LOD) principles [1], semantic recommender systems and applications based on linked datasets become an important research area. Semantic datasets, which represent knowledge as a huge network of nodes and labeled edges, provide the basis for the effective deployment of (natural) language independent knowledge processing. Thus, the approach for processing semantic datasets abstracts from classical text processing tasks (e. g., handling of synonyms, homonyms, typos, multi-lingual content, ambiguous terms), but focuses on deploying the relationship between unique entities. Moreover, the ontology based semantic representation of data simplifies the reuse of existing datasets and the integration of new information sources.

For many domains (such as music, movies, and geographic locations), large semantic encyclopedic datasets are available from Freebase [2] and DBpedia [3]. These encyclopedic datasets provide generally accepted, almost static knowledge. The data is represented as nodes ("vertexes") connected by labeled edges, describing the relationship between the nodes. The entities (such as artists, events, locations, or points of interest) represented as nodes are usually annotated with meta-data (such as images or labels for different languages).

An important question, when working with semantic datasets, is how to discover the entities (of a specific type) most closely related to a set of input entities. The computation of related entities is used for interfering knowledge for enriching profiles or for calculating recommendations. The main questions that have to be answered when calculating related entities are:

1) What types of edges should be considered for computing the semantic similarity between nodes?
2) How to assign weights to labeled edges?
3) How to combine edge weights of paths between the source node and the destination node?
4) How to efficiently compute related items based on huge datasets? Which network models adequately reduce the complexity without spoiling the result quality?

In this paper we discuss and compare several algorithms for computing the most-related entities for a weighted set of input entities. The evaluation is based on a recommender system for the music domain. In contrast to most existing systems that focus on user ratings and user generated tags, our system bases on well accepted encyclopedic data. Thus, we concentrate on computing related entities and not on personalized recommendation (personalized recommendation cannot be found in an encyclopedia). The computation of related entities based on encyclopedic data has the advantage that results are built on a reliable dataset and thus are suitable for enriching sparse user profiles.

The paper is structured as follows: Section II gives an overview of related work; Section III explains the dataset used for evaluating our approach. In Section IV, we introduce our approach in detail. Section V presents a recommender systems implemented based on our approach. The experiments and the evaluation performed for proving the properties of our approach are discussed in Section VI. Finally, a conclusion and an outlook to future work are given in Section VII.

 ISBN: 978-1-61208-175-5

II. Related work

Most of the existing recommender systems apply collaborative filtering (CF) methods [4], [5], [6]. Recommendations are calculated by analyzing the similarity of user profiles (user-based CF) or the similarity of rated items, such as artists, albums, films, books (item-based CF). Some authors [7], [8], [9] combine user-based CF and item-based CF approaches. These hybrid recommender systems often deploy expert-defined, domain-specific rules for a scenario dependent combination of different feature types.

For the entertainment domain several recommender systems exist, such as the FOAFing-the-music project [10], combining social networks and user ratings. Another active research area is the use of *Linked Open Data* [11]. Comprehensive ontologies have been defined for the semantic storage of knowledge for the music domain. Well-known ontologies are provided by the Music Ontology project [12] and the Music Similarity Ontology project [13]. These ontologies focus on the aggregation of various data sources and on providing fine-grained semantic descriptions of relevant entities.

III. Dataset

We use an encyclopedic dataset retrieved from Freebase as data source for testing our semantic processing framework. For the evaluation we use a rating dataset retrieved from LastFM (http://www.last.fm/). Freebase is a comprehensive data source for semantic data containing information about almost every domain. In our scenario (computing the most-related entities in the music domain) we make use of a subset of the data retrieved from Freebase consisting of the four entity types *Artists, Albums, Tracks*, and *Genres*. The relationship between *Artists* and *Genres* describes the genre in which an artist usually works; the relationship between *Albums* and *Artists* describes the album releases of each artist, and finally the relationship between *Albums* and *Genres* defines a genre assignment for each album. The created dataset is schematically visualized in Figure 1.

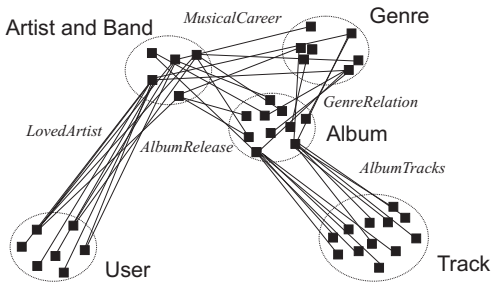


Figure 1. The semantic music dataset.

To compare an encyclopedic "recommender" with a rating-based recommender, we interlink the encyclopedic dataset retrieved from Freebase with a rating dataset retrieved from LastFM (collected in December 2010) consisting of 40,000 user profiles. The linkage of the datasets had been established based on the artist names and the MusicBrainz ID [14]. The size of the respective entity sets and relationship sets is shown in Table I.

Table I
The number of entities and edges in the encyclopedic dataset.

	# entities	# edges			
		Artists	Genre	Albums	Tracks
Artists	417217	-	79543	374445	-
Genre	3082	79543	-	90444	-
Albums	438180	374445	90444	-	1048565
Tracks	1048576	-	-	1048565	-

IV. Approach

The necessary steps for computing the most-related entities for a set of input items are: Assign numerical edge weights (describing the similarity between entities) based on the edge labels, and define rules ("an algebra") describing how to combine the edge weights. Additionally, models for coping with the network complexity must be defined, speeding up the computation process and reducing the noise present in real-world datasets.

We discuss the challenges and solutions for each step in detail in the following paragraphs. At first, we analyze the task of link prediction in a semantic network. In other words, we infer for a given node the entities strongly related and suggest to add edges to these nodes [15], [16]. In our application scenario, the prediction of new edges means to compute the most-related entities for a given input entity that are not directly connected by an edge in the semantic dataset. We focus on algorithms allowing us to provide explanations for each predicted entity. In many scenarios this is important since good explanations help to increase the user's trust and confidence in the recommendations as wells as in the recommender system itself [17].

How to define relatedness: For computing related items in a large semantic network, we have to define criteria for measuring the semantic *similarity* between two entities. Criteria for defining the similarity between two nodes in a semantic network are:

- Entities connected by a short path are more related to each other than entities connected by a long path.
- Entities connected by several different parallel paths are more closely related than entities connected by one path only.
- The edge labels (and the derived edge weights) of a path between two nodes should influence the computed node relatedness. In general, the edge weight might depend on the path context (in other words, on the other edges of a path).

 ISBN: 978-1-61208-175-5

Path Algebra: Based on the proposed criteria for the relatedness of nodes of a network, an edge algebra is defined. Well-known approaches for combining the edge weights of a path are the *shorted path distance*, the *resistance distance*, and the *weighted path distance* [18]. The rules for calculating the path weight according to the different combination approaches are shown in Table II.

Table II
THE TABLE SHOWS THE FORMULAS FOR CALCULATING THE PATH WEIGHTS FOR (A) PARALLEL EDGES AND (B) FOR A SEQUENCE OF EDGES. THE DISCOUNT FACTOR γ ENSURES THAT SHORT PATHS GET A HIGHER WEIGHTING THAN LONG PATHS.

		Weighted Path	Resistance Distance	Shortest Path
(A)	w_0 w_1 ⋮ w_n	$w = \sum_{i=0}^{n} w_i$	$w = \frac{1}{\sum_{i=0}^{n} w_i}$	$w = \min_{i=0}^{n} w_i$
(B)	w_0 w_1 ⋯ w_n	$w = \sum_{i=0}^{n} w_i$	$w = \sum_{i=0}^{n} w_i$	$w = \sum_{i=0}^{n} w_i$

Computing recommendations on semantic datasets: Large semantic datasets usually consist of several node types (often annotated with `rdf:type`) and edge sets connecting exactly two entity sets (*bipartite relationship sets*). Additionally, unipartite relationships, connecting nodes within one entity set, may exist (e. g., to model hierarchies of entities). Each relationship between the entity sets has a semantic meaning that can be used for deriving edge weights. In general, two entity sets can be connected by several different relationship sets, describing different semantic associations.

For computing the most-related items for a set of input entities, we define which relations can be combined to build valid paths. In other words, we identify a set of valid pipes, describing the edge types combined in a path as well as the minimal and maximal path length. This approach allows us to assign edge weights based on the context of an edge. Thus, we do not use a static edge weight, but choose the edge weight dependent on the semantic meaning of an edge in a path. Moreover, for each relationship type specific models can be defined allowing us to consider the special properties of each relationship type.

Memory-based Recommender: To compute related entities for a given set of input items, we determine the entities best connected to the input entities (according to the defined edge algebra). We implement the search based on a Branch and Bound algorithm [19], adapted to handle parallel paths in the search process. The search process takes into account the different semantic edge types and ensures that only paths consisting of valid edge sequences are considered. The advantage of path-based recommenders is that no additional effort is needed for building a dataset model. Thus, updates in the dataset immediately affect the computed results. Another advantage of calculating the most-related nodes directly on the dataset is that the computed paths can be used as explanations for the derived nodes. In most scenarios the path length is limited so that the explanations are not too complex ensuring that users understand these computed explanations. An example for an path-based explanation (taken from the encyclopedic recommender system for the music domain) is shown in Figure 2. Starting from the input node `Kelis`, the path recommender used five genre nodes, to find several parallel paths to the artist `Pink`. Edge weights and edge labels are not shown in the explanation graph in order to keep the explanation simple.

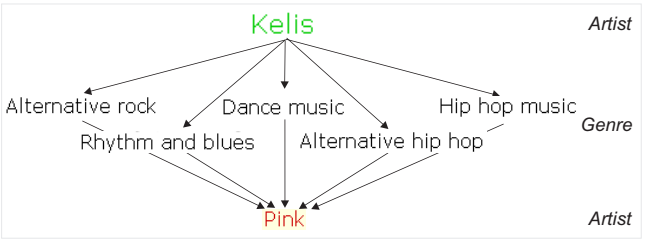


Figure 2. Explanation of a path-based recommendation (used in our music recommendation web application). The user can see the different nodes that are relevant for recommending the artist *Pink*.

Model-based Recommender: While working with real-world data, semantic relationship sets are often huge, noisy, and sparse. Models for simplifying the semantic relationship set are applied to cope with these problems.

Clustering: An efficient approach for reducing the entity set size and the relationship size is aggregating similar entities into clusters. The advantage of this approach is that most users understand the idea of clustering and path-based explanations can be computed (handling clusters as "virtual" entities). Figure 3 shows an example for an explanation containing clustered entity sets.

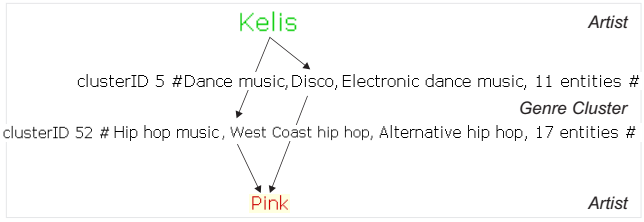


Figure 3. Explanation of a path-based recommendation using a clustered entity set. By aggregating similar entities into clusters, the graph complexity and thus the complexity of the provided explanation are reduced.

Analyzed approaches for clustering: We focus on Hierarchical Agglomerative Clustering (HAC) [20]. The advantage of HAC is that the desired number of clusters does not need to be chosen in advance. The distance measure used for clustering may take into account several different entity properties. In our music recommendation scenario, we used a similarity measure based on a weighted combination of the

 ISBN: 978-1-61208-175-5

genre name similarity and user-defined genre hierarchy data (retrieved from Freebase) for clustering the music genres.

Low Rank Approximation: An alternative approach for reducing the complexity is to compute a low rank approximation of the adjacency matrix for a relationship set. For this purpose we calculate the singular value decomposition (SVD) of the normalized adjacency matrix A and consider only the first k latent dimensions.

$$A = USV^T \approxeq U_k S_k V_k^T$$

The adjacency matrix A is decomposed into a diagonal matrix S, containing the singular values of A in descending order. The matrices U and V consist of the left-singular and right-singular vectors for S. The low rank approximation of A considers only the largest k singular values of A and the respective eigenvectors (U_k, V_k^T).

The advantage of this approach is, that it allows us an efficient reduction of the matrix size. Moreover, the low rank approximation has been proven to be a good model for large sparse matrices [21]. Disadvantages of this approach are on the one hand that no easily understandable explanations can be provided and on the other hand that the singular value decompositions is resource-demanding. Dataset updates require a recalculation of the matrix decomposition.

Conclusion: In this section we discussed the problem of computing related items for a given set of entities considering the node and edge semantics. In contrast to most of the existing systems which consider only one edge type (typically "like" or "is similar to") our system focuses on analyzing the edge semantics. The combination of heterogeneous edges takes into account the semantics of respective paths. We explained different approaches for combining edge sequences and parallel paths (*edge algebra*) dependent on the respective node types and edge labels. A promising approach consists of expert-defined rules, reflecting the specific properties of the respective domain, and optimized parameter settings computed using machine learning methods based on the available training data.

Additionally, we discussed the advantages and disadvantages of memory-based and model-based approaches for efficiently computing related entities. The analysis showed that memory-based approaches allow providing user-understandable explanations without precomputing sophisticated models. Model-based approaches allow reducing the complexity and taking into account the noise in real-world datasets.

V. Implementing a system for encyclopedic music recommendations

Based on the developed framework for semantic data processing, we implemented a web application for suggesting entities semantically related to the entities present in the user profile. As the knowledge base for our recommender system, we use a semantic dataset for the music domain retrieved from Freebase (see Table I).

User profile: The user preferences are stored as a set of weighted entities. The entities define artists, genres, tracks and albums the user "likes". User preferences are collected implicitly (by analyzing the user behavior) and explicitly (allowing the user to enter entities she is interested in). A disambiguation component computes potentially matching entities to the user's input ensuring that only valid entities are added to the user profile. The disambiguation component is needed due to the fact that a user-entered name may represent different entities. For instance, the name `Madonna` may stand for an American singer, her first album or the second studio album from the American band `...And You Will Know Us by the Trail of Dead`.

The analyzed edge combinations: For computing the recommendations based on the encyclopedic dataset, we tested which semantic relationship sets should by combined to provide good results. We focus on path of limited length (maximal 4 edges) consisting of edges from only one edge set, since the meaning of those paths is understood best by the users. While calculating the most-related entities for a set of user profile entities several different relationship sets are taken into account. Figure 4 shows an example for computing related items for the entities `Dr. Dre` and `50 Cent`. The entity `Eminem` is related to the input entities because `Eminem` has four music genres in common with `Dr. Dre` and `50 Cent`. Moreover, he worked together at the albums `Welcome To The Dogg House`, `The Slim Shady LP` and `Up In Smoke Tour`.

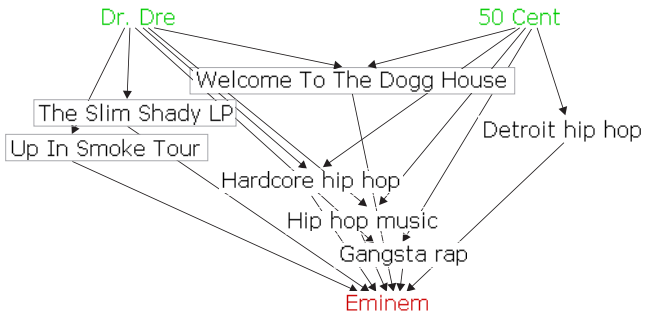


Figure 4. The figure visualizes the considered path of length 2 between user profile entities and the derived entity `Eminem`. Each path consists of edges from only one relationship set.

Since a joint album usually implies a close similarity between two artists, in our web application the paths based on the Artist-Album relationship have a higher weight than paths based on the Artist-Genre relationship. Only in the case that no related entities can be computed, neither based on the Artist-Album relationship nor based on the Artist-Genre relationship, more complex paths (such as Artist → Genre → Album → Artist) are taken into account.

 ISBN: 978-1-61208-175-5

Preliminary experiences: The first evaluation results of the developed encyclopedic "recommender" system showed, that the entities calculated to be related to the user profile are useful to the user. The huge number of nodes enables the system to compute results even for only locally known artists. In contrast to systems focused on individual ratings, the suggested entities are related to the user profile (according to the encyclopedic knowledge base) and not based on the user's taste. Most users liked the idea of providing explanations for the results, especially if a recommendation is not obviously related to the user interest. The presentation of explanations as a graph seems to be an acceptable solution as long as the explanations are not too complex. Hence, we simplify complex explanation graphs keeping only the edges with the highest weights.

VI. Experiments and Evaluation

To evaluate the implemented algorithms, we analyze different scenarios.

A. Link prediction on encyclopedic data

We analyze the task of predicting links on the encyclopedic dataset retrieved from Freebase. We focus on the scenario of computing related artists for a given set of artists (e. g., for the entities from a user's preference profile). Following the idea of cross-validation, we split the edge set of our dataset into a training set and a test set. Entities connected with less than two edges are not considered in the evaluation. Based on the edges of the training set, the recommender component predicts edges to the most-strongly connected entities and provides a list of edges ordered by the semantic similarity between the connected nodes. The prediction precision is evaluated with the test set. Since the number of entities related to the input entity set varies over the user profiles, the Mean Average Precision (MAP) [22] is used as performance measure. The MAP for a set of user profiles $P = \{p_1, p_2, \ldots, p_n\}$ is calculated as follows:
Let $\mathrm{Prec@}i(L_p)$ be the Precision of the first i items in the predicted result list L for the profile $p \in P$, and $\mathrm{rel@}i(R_p)$ be a Boolean function returning 1 if the ith item in the list L is relevant, then

$$\mathrm{MAP}(P) = \frac{1}{|P|} * \sum_{p \in P} \sum_{i=1}^{m} \mathrm{Prec@}i(L_p) \cdot \mathrm{rel@}i(L_p)$$

Memory-based Recommenders: We analyze the task of predicting related entities directly on the semantic graph retrieved from Freebase (see Figure 1). For the evaluation we performed the following steps: (1) We randomly select a node. (2) The set of edges connected to this node is split into a training set and a test set (50% / 50%). (3) Based on the training set we compute the most-related nodes limiting the maximal considered path length. (4) The predicted nodes are evaluated with the test set. (5) We calculate the average over all the evaluation results for 10,000 nodes. Figure 5 shows the observed prediction precision for the two baseline strategies (predict edges to randomly chosen entities, and predict edges to the entities having the highest number of edges) and for the path-based recommender considering a maximal path length of two or four respectively. The results show that our approach provides highly relevant prediction results. A higher search depth (4 instead of 2) leads to slightly improved results as more nodes are taken into account. The high prediction precision can be explained by the fact, that in the deployed music dataset several parallel paths for connecting two entities exist. Moreover, the artists in the music dataset seem to form "clusters" whose nodes are well connected but have only a small number of connections to other entities.

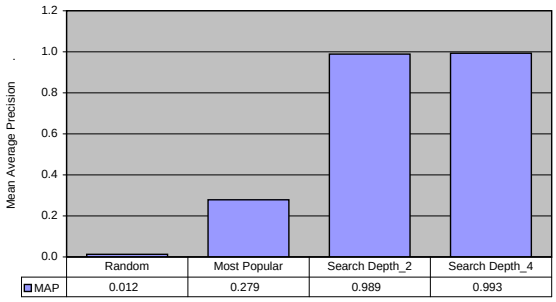


Figure 5. The evaluation of link prediction for artists based on the Freebase dataset.

Link prediction on the clustered entity sets: Due to the large number of music genres in the used Freebase dataset we apply a clustering algorithm for aggregating similar genres. We analyze how the edge prediction precision depends on the number of clusters. The clusters are computed based on a hierarchical agglomerative clustering algorithm. For calculating the distance between two music genres we determine the number of artists and albums directly connected with theses genres. Additionally, we consider the metadata from Freebase describing relations between the music genres.

In our evaluation we compute clusters for the genre entity set and apply a path-based search algorithm with a search depth of two. The measured results (see Figure 6) show that aggregation of the 15% most similar genres into clusters leads to only a minimal decrease of the precision. In the case of a small number of clusters the precision decreases appreciably. For the analyzed scenario the use of ≈ 900 clusters provides reasonable results while reducing the considered genre entity set size by $\approx 15\%$, and thus reducing the complexity of the dataset.

B. Profile enrichment based on encyclopedic data

We interlink the encyclopedic music dataset from Freebase with LastFM user profiles and analyze how our recommender improves the collaborative computation of recommendation results by enriching small user profiles.

 ISBN: 978-1-61208-175-5

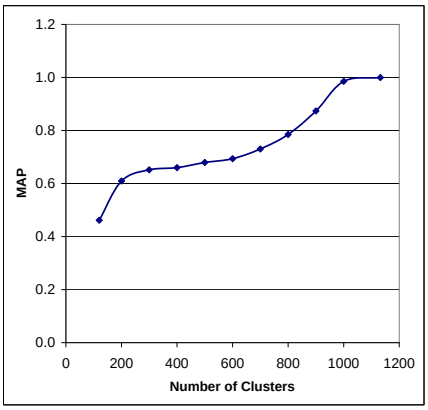


Figure 6. The evaluation of link prediction for artist based on the Freebase dataset using clustered music genres.

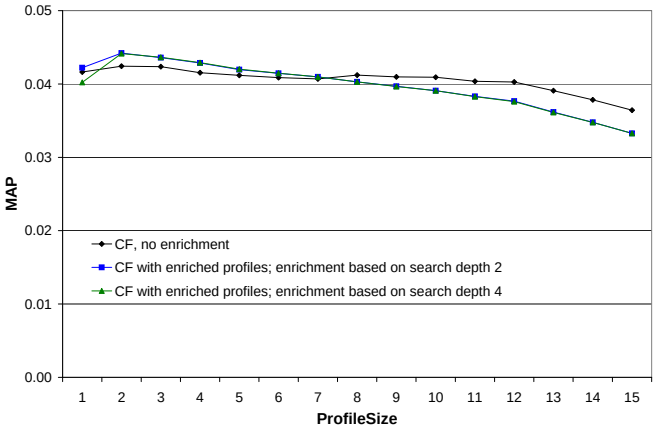


Figure 7. The evaluation shows that profile enrichment based on encyclopedic knowledge improves the precision of collaborative filtering for users with a small profile. For users with more than six entries the profile enrichment reduces the recommendation precision.

For the evaluation, we use 10,000 LastFM user profiles having at least 30 (to have enough information for a proper evaluation) and at most 50 preferred artists. We split each user profile into a training set containing $n \quad (1 \leq n \leq 10)$ artists and a test set containing the remaining artists. As a baseline for our evaluation, we use a standard collaborative filtering (CF) algorithm, computing the similarity between two users based on the number of common entities. CF computes the 100 most similar users (based on the number of common artists) and predicts the entities most frequently present in these profiles. While determining similar users, only the training set for the user (for which the recommendations are computed) is taken into account. The recommendation precision is calculated based on the test set.

We analyze how the recommendation performance changes, if we enrich user profiles using the encyclopedic data retrieved from Freebase. For the recommender on the encyclopedic dataset we consider the artist-genre relation and search depths of two and four. Figure 7 shows that profile enrichment improves the recommendation precision for small user profiles. For users having more than $\approx$ 7 profile entries the profile enhancement leads to less precise results. Thus, encyclopedic knowledge helps to improve the recommendation results for new user. If the user profile consists of an adequate number of entities a profile enrichment based on encyclopedic data should not be applied.

The results can be explained by the fact that similar users cannot be reliably computed for users with a very small profile. Thus, enriching the profile with related entities improves the calculation of similar users and the computation of predictions. Due to the fact that encyclopedic knowledge does not consider the individual user taste, the profile enrichment adds fuzziness to the profile. For large user profiles the items (added by the enrichment) adulterate the user profiles resulting in less precise recommendation results.

VII. Conclusion and Future Work

In this paper, we introduced a new semantic recommender framework and discussed different algorithms for the efficient processing of large semantic datasets. We explained our graph-based recommendation approach utilizing model- and memory-based link prediction methods. We showed how to provide explanations to increase the trust in the computed recommendations. With the aggregation ("clustering") of similar entities we could reduce the computational complexity with the trade-off of a small loss of precision. The evaluation of the link prediction approach shows that our recommender provides precise link prediction results on the encyclopedic dataset. The analyzed algorithms require only limited resources and provide comprehensible explanation for the recommendations.

We also demonstrated the application of our recommender to enrich user profiles and explained how the enhanced profiles can be used to improve collaborative filtering. The results showed that encyclopedic data helps only in the case of very small user profiles. This can be explained by the fact that for a user having a small user profile users with similar interests cannot be reliably computed. A profile enrichment based on encyclopedic data improves the computation of similar users and leads to better recommendations. Thus, the profile enrichment helps to overcome the cold-start problem [23]. For users with a big profile encyclopedic data does not improve the recommendation precision. A reason for this is that our encyclopedic data neither considers individual user preferences nor the "quality" of albums or musicians.

Future Work: As future work, we want to analyze and integrate additional recommender models based on matrix decomposition [24], [25] and graph kernels [26]. Preliminary tests with these methods show promising results in effectively reducing the dataset complexity and reducing the noise in the datasets. Furthermore, it is intended to extend the

 ISBN: 978-1-61208-175-5

dataset with additional entities and meta-information. First, we want to extend the scope of the music recommendation scenario by adding information such as movies and actors to test our approach in a cross-domain recommendation scenario. Second, we want to add meta-information to the encyclopedic dataset like "quality" of a node to extend the recommendation approach with methods that do not only take into account the graph structure but also the type and quality of a node. Such quality information can be the popularity of an artist or the commercial success. Ongoing work is the preparation of a user study where we want to get real user feedback about the recommendation and explanation quality in order to validate our results based on the automatic evaluation.

REFERENCES

[1] Bizer, C., Heath, T., Idehen, K., Berners-Lee, T.: Linked data on the web (ldow2008). In: Proceedings of the 17th International Conference on World Wide Web. WWW '08, New York, NY, USA, ACM (2008) 1265–1266

[2] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proc. of the 2008 ACM SIGMOD intl. conf. on Management of data. SIGMOD '08, New York, NY, USA, ACM (2008) 1247–1250

[3] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In Aberer, K., Choi, K.S., Noy, N., Allemang, D., Lee, K.I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudr-Mauroux, P., eds.: The Semantic Web. Volume 4825 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2007) 722–735

[4] Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the International Conference on Research and Development in Information Retrieval. (1999) 230–237

[5] Sahoo, N., Krishnan, R., Duncan, G., Callan, J.: Collaborative filtering with multi-component rating for recommender systems. In: Proceedings of the Workshop on Information Technologies and Systems. (2006)

[6] Sun, J., Zeng, H., Liu, H., Lu, Y., Chen, Z.: CubeSVD: A novel approach to personalized web search. In: Proc. of the Intl. World Wide Web Conference. (2005) 382–390

[7] Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Transactions on Knowledge and Data Engineering **17** (2005) 734–749

[8] Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Transactions Information Systems **23**(1) (2005) 103–145

[9] Zhou, D., Orshanskiy, S., Zha, H., Giles, C.: Co-ranking authors and documents in a heterogeneous network. In: Proc. of the Intl. Conf. on Data Mining. (2007) 739–744

[10] Celma, O.: Foafing the music: A music recommendation system based on rss feeds and user preferences. In: Proc. of the 6th Intl. Conf. on Music Information Retrieval 2005. (2005) 464–467

[11] Hausenblas, M.: Exploiting linked data to build web applications. IEEE Internet Computing **13** (2009) 68–73

[12] Yves, R., Samer, A., Mark, S., Frederick, G.: The music ontology. In: Proc. of the Intl. Conf. on Music Information Retrieval. ISMIR 2007 (2007) 417–422

[13] Jacobson, K., Raimond, Y., Gängler, T.: The similarity ontology - musim. Technical report, School of EECS, Queen Mary, University of London (2010) http://kakapo.dcs.qmul.ac.uk/ontology/musim/0.2/musim.html.

[14] Swartz, A.: Musicbrainz: a semantic web service. Intelligent Systems, IEEE **17**(1) (jan/feb 2002) 76 – 77

[15] Popescul, A., Ungar, L.H.: Statistical relational learning for link prediction. In: Proceedings of the Workshop on Learning Statistical Models from Relational Data. (2003)

[16] Taskar, B., Wong, M.F., Abbeel, P., Koller, D.: Link prediction in relational data. In: Proceedings of Neural Information Processing Systems. (2004)

[17] Swearingen, K., Sinha, R.: Beyond algorithms: An hci perspective on recommender systems. In: ACM SIGIR Workshop on Recommender Systems, New Orleans, LA, USA (2001)

[18] Tizghadam, A., Leon-Garcia, A.: Betweenness centrality and resistance distance in communication networks. Network, IEEE **24**(6) (november-december 2010) 10 –16

[19] Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. 2 edn. Pearson Education (2003)

[20] Zhao, Y., Karypis, G.: Evaluation of hierarchical clustering algorithms for document datasets. In: Proc. of the 11th Intl. Conf. on Information and Knowledge Management. CIKM '02, New York, NY, USA, ACM (2002) 515–524

[21] Kunegis, J., Lommatzsch, A.: Learning spectral graph transformations for link prediction. In: ICML '09: Proceedings of the 26th Annual Intl. Conf. on Machine Learning, New York, NY, USA, ACM (2009) 1–8

[22] Voorhees, E.M., Harman, D.K., eds.: TREC: Experiment and Evaluation in Information Retrieval. MIT Press (2005)

[23] Zhang, Z.K., Liu, C., Zhang, Y.C., Zhou, T.: Solving the cold-start problem in recommender systems with social tags. EPL (Europhysics Letters) **92**(2) (2010) 28002

[24] Saul, L.K., Weinberger, K.Q., Sha, F., Ham, J., Lee, D.D.: Spectral Methods for Dimensionality Reduction. In: Semi-supervised Learning. MIT Press (2006)

[25] Lathauwer, L.D., Moor, B.D., Vandewalle, J.: A multilinear singular value decomposition. Matrix Analysis and Applications **21**(4) (2000) 1253–1278

[26] Kunegis, J., Lommatzsch, A., Bauckhage, C., Albayrak, S.: On the scalability of graph kernels applied to collaborative recommenders. In: Proceedings ECAI 2008, Workshop on Recommender Systems. (2008) 35–38

 ISBN: 978-1-61208-175-5

SPARQL Query Processing Using Bobox Framework

Miroslav Čermák, Zbyněk Falt, Jiří Dokulil and Filip Zavoral
Charles University in Prague, Czech Republic
{*cermak,falt,dokulil,zavoral*}*@ksi.mff.cuni.cz*

***Abstract*—Proliferation of RDF data on the Web creates a need for systems that are not only capable of querying them, but also capable of scaling efficiently with the growing size of the data. Parallelization is one of the ways of achieving this goal. There is also room for optimization in RDF processing to reduce the gap between RDF and relational data processing. SPARQL is a popular RDF query language; however current engines do not fully benefit from parallelization potential. We present a solution that makes use of the Bobox platform, which was designed to support development of data-intensive parallel computations as a powerful tool for querying RDF data stores. A key part of the solution is a SPARQL complier and execution plan optimizer, which were tailored specifically to work with the Bobox parallel framework. The performance of the system is compared to the Sesame SPARQL engine.**

Keywords-SPARQL; Bobox; query optimization.

I. Introduction

SPARQL [1] is a popular RDF (Resource Definition Framework) query language. It contains capabilities for querying graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be result sets or RDF graphs.

The Bobox framework was designed to support development of data-intensive parallel computations [2], [3]. The main idea behind Bobox is to divide a large task into many simple tasks that can be arranged into a non-linear pipeline. These simple tasks are performed by *boxes*. They are executed in parallel and the execution is driven by the availability of data on their inputs. The developer of such boxes does not have to be concerned about problems such as synchronization, scheduling and race conditions. All this is done by Bobox itself. The system can easily be used as a database execution engine; however, each query language requires its own front-end that translates a request (query) into a definition of the structure of the pipeline that corresponds to the query.

In the paper, we present a way in which we used the Bobox framework to create a tool for effective parallel querying of RDF data [4] using SPARQL. The data are stored using an in-memory triple store which consists of one three-column table and a set of indexes. We provide a brief description of query processing using SPARQL-specific parts of the Bobox and provide results of benchmarks. Benchmarks were performed using the SP^{2}Bench [5] query set and data generator.

The rest of the paper is structured as follows: Sections II and III describe the Bobox framework and models used to represent queries during their processing. Section IV contains a description of the SPARQL compiler and steps performed during query processing. Bobox back-end processing and SPARQL specific boxes are discussed in the Section IV-D. Section V presents our experiments and a discussion of their results. Section VI describes future directions of research and concludes the paper.

II. Bobox Framework

A. Bobox Architecture

The Bobox parallelization framework has two primary goals: to simplify writing parallel, data intensive programs and to serve as a testbed for the development of generic parallel algorithms and data-oriented parallel algorithms. The main aspects that make writing parallel programs easier include the following: all synchronization is hidden from the user; most technical details (NUMA, cache hierarchy, CPU architecture) are handled by the framework; high-performance messaging is the only means of communication and synchronization; and it is built around easy-to-comprehend basic paradigms such as task parallelism and non-linear pipeline.

Bobox provides a run-time environment that is used to execute a non-linear pipeline in parallel. The pipeline consists of computational components provided by the user and connecting parts that are part of the framework. The structure of the pipeline is defined by the user, but the communication and execution of individual parts is handled by the run-time; a component is executed when it has data waiting to be processed on its inputs. This simplifies the design of the individual computational components, since communication, synchronization and scheduling are handled by the framework.

Compared to scientific workflows, the Bobox boxes are usually smaller than actors or other workflow elements and they never encapsulate user interaction or unreliable remote communication.

B. Task Level Parallelism

The environment with many simple components and pipeline-based communication is very suitable for task level parallelization. In this paradigm, the program is not viewed as a process divided into several threads. Instead, it is seen

 ISBN: 978-1-61208-175-5

as a set of small tasks. A *task* is a piece of data together with the code that should be executed on the data. Their execution is handled by a *task scheduler*. The scheduler maintains a pool of tasks to be executed and a pool of execution threads and allocates the tasks to the threads. At any given time, a thread can either be executing a task or be *idle*. If it is idle, the task scheduler finds a suitable task in the task pool and starts the execution of the task on the idle thread.

C. Run-time Architecture

One of the main differences between other parallelization frameworks and the Bobox architecture is the way the user's code interacts with Bobox. OpenMP [6] and TBB [7] are used to invoke parts of the code in parallel; MPI [8] provides means for communication between processes. Bobox is more similar to the first two systems; however, there are two key differences. First, it uses a declarative approach to describe the way in which elements of the computation are put together. Second, it provides more services to the user code (data transport, flow control etc.), but also imposes greater restrictions (only pipeline, no recursive calls, etc.).

The parallel execution environment is somewhat similar to that of TBB, since it contains a task pool and several threads that execute tasks from that pool. However, the way in which the tasks are created and added to the pool is completely different [9]. In TBB, this is controlled either directly by the user's code or by using a thin layer of parallel algorithms provided by the library.

In Bobox, the user first specifies a *model*. The model defines the way in which the individual computational components are connected. The model is then *instantiated* to produce a *model instance*. The elements of the model instance are used as tasks. When they are ready, they are *enqueued* – added to the task pool. Later, a thread takes a task from the pool, performs the action (*invokes* the task) and then the model instance element is returned and can be used again as a new task and added to the pool.

D. Scheduling

The Bobox system is well suited for a certain class of problems, due to the way in which the system decides what computational components should be executed. This is controlled by the flow of the data through the pipeline. The data must be passed in a way defined by the system, so that the system is aware of the fact that a component consumed or created some data. This simplifies the design of the individual computational components; they do not have to be concerned with controlling the execution and data flow.

The basic Bobox computational component is a *Box*. Boxes are used for the implementation of basic operations such as joins (see Section IV-D for a more details).

III. Query Representation

During query processing, our SPARQL compiler uses different representations of the query itself. They are chosen according to the needs of each processing step. In the following sections, we mention models used during query rewriting and generation of execution plan.

A. SQGM Model

Pirahesh et al. [10] proposed the Query Graph Model (QGM) to represent SQL queries. Hartig and Reese [11] modified this model to represent SPARQL queries (SQGM). With appropriate operations definition, this model can be easily transformed into Bobox pipeline definition, so it was ideal candidate to use.

SQGM model can be interpreted as a directed graph (in our case a directed tree). Nodes represent operators and are depicted as boxes containing headers, body and annotations. Edges represent data flow and are depicted as arrows that follow the direction of the data. Figure 1 shows an example of a simple query represented in the SQGM model.

This model is created during execution plan generation step and is used as a definition for the Bobox pipeline.

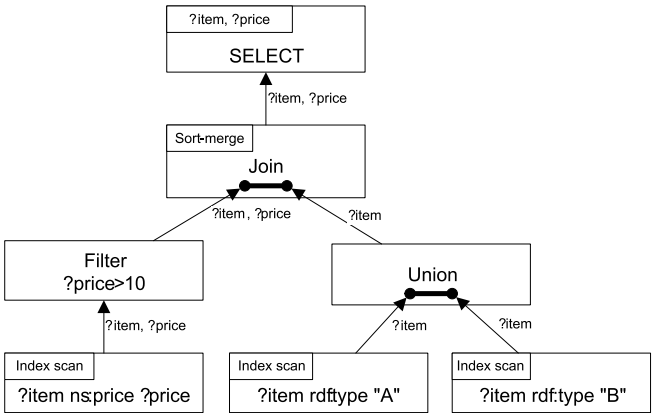


Figure 1. Example of SQGM model.

B. SQGPM Model

In [12], we proposed the SPARQL Query Graph Pattern Model (SQGPM) as the model that represents query during optimization steps. This model is focused on representation of the SPARQL query graph patterns [1] rather than on the operations themselves as in the SQGM. It is used to describe relations between group graph patterns (graph patterns consisting of other simple or group graph patterns). The ordering among the graph patterns inside a group graph pattern (or where it is not necessary in order to preserve query equivalency) is undefined. An example of the SQGPM model graphical representation is shown in Figure 2.

Each node in the model represents one group graph pattern that contains an unordered list of references to graph patterns. If the referenced graph pattern is a group graph pattern, then it is represented as another SQGPM node. Otherwise the graph pattern is represented by a leaf.

 ISBN: 978-1-61208-175-5

The SQGPM model is built during the syntactical analysis and is modified during the query rewriting step. It is also used as a source model during building the SQGM model.

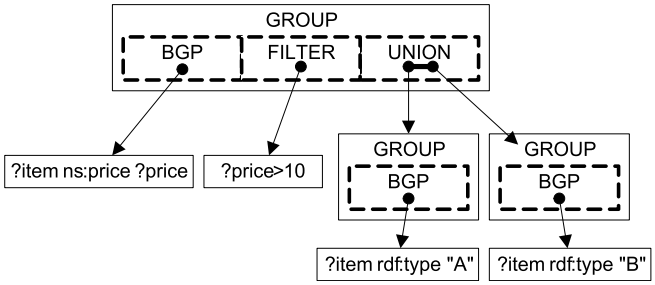


Figure 2. Example of SQGPM model.

IV. Query Processing

Query processing is performed in a few steps by separate modules of the application as shown in Figure 3. First steps are performed by the SPARQL front-end represented by compiler. The main goal of these steps is to validate the compiled query, pre-process it and prepare the optimal execution plan according to several heuristics. Execution itself is done by the Bobox back-end where execution pipeline is initialized according to the plan from the front-end. Following sections describe steps done by the compiler in a more detail way.

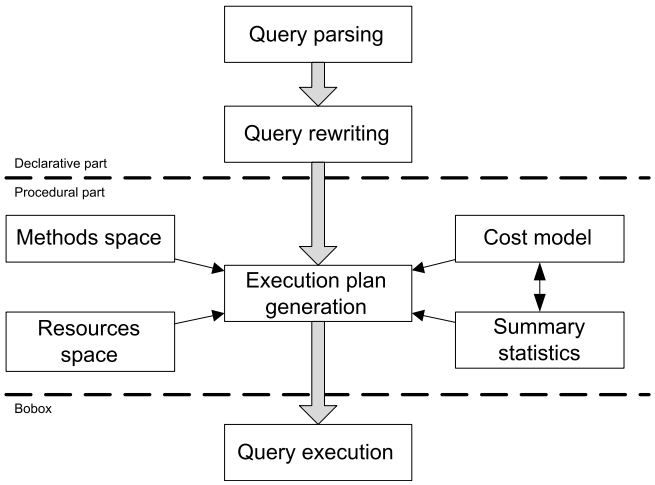


Figure 3. Query processing scheme.

A. Query Parsing

The query parsing step uses standard methods to perform syntactic and lexical analysis according to W3C recommendation. The input stream is transformed into a SQGPM model. Transformation also includes expanding short forms in query, replacing aliases and transformation of blank nodes into variables.

B. Query Rewriting

The second step is query rewriting. We cannot expect that all queries are written optimally (they may contain duplicities, constant expressions, inefficient conditions, redundancies etc.). So, the goal of this phase is to normalize queries to achieve a better final performance. We use the following operations:

- Merging of nested *Group graph patterns*
- Duplicities removal
- *Filter*, *Distinct* and *reduced* propagation
- Projection of variables

It is also necessary to check applicability of each operation with regards to the SPARQL semantics, before it is used to preserve query equivalency. Query representation is the same as in the previous step.

C. Execution Plan Generation

In the previous steps, we described some query transformations that resulted in a SQGPM model. However this model does not specify complete order of all operations. Main goal of the execution plan generation step is to transform the SQGPM model into an execution plan. This includes selecting from different join operation orderings, join types and selecting the best strategy to access the data stored in the physical store.

The query execution plan is built from the bottom to the top using dynamic programming to search part of the search space of all possible joins. This strategy is applied to each group graph pattern separately because the order of the patterns is fixed in the SQGPM model. Also, the result ordering is considered, because a partial plan that seems to be worse locally, but produces a useful ordering of the result may provide a better overall plan later. The list of available atomic operations (e.g., the different types of joins) and their properties are provided by the *Methods Space* module.

In order to compare two execution plans, it is necessary to estimate the *cost* of both plans – an abstract value that represents the projected cost of execution of a plan on the actual data. This is done with the help of the *cost model* that holds information about atomic operation efficiency and *summary statistics* gathered about the stored RDF data.

Search space of all execution plans could be extremely large, so we used heuristics to reduce the complexity of the search. At first, only left-deep trees of join operations are considered. This means that right operand of join operation may not be another join operation. There is one exception to this rule – avoiding cartesian products. If there is no other way to add another join operation without creating cartesian product, the rest of the unused operations is used to build separate tree recursively (using the same algorithm) and result is joined with the already built tree. This modification greatly improves plans for some of the queries we have tested and often significantly reduces the depth of the tree.

 ISBN: 978-1-61208-175-5

The final execution plan is represented using SQGM model and later transformed int a Bobox model. This transformation is completely straightforward.

D. Query Representation for Back-end

After the execution plan is generated, it is transformed into a serialized form and passed to the back-end. The back-end deserializes the plan and instantiates boxes provided by the runtime implementation. Boxes are connected according to the plan and computation may then be started. The serialization and deserialization is useful since it provides many benefits, such as:

- When distributed computation support is added, text representation is safer than (e.g., binary), where problems with different formats, encodings or reference types may appear.
- Serialization language has very simple and effective syntax; serialization and deserialization is much faster than (e.g.) the use of XML.
- Text representation is independent on the programming language; new compilers can be implemented in a different language.
- Compilers can generate plans that contain boxes that have not yet been implemented, which allows for earlier testing of the compiler during the development process.

E. Runtime

Another important part of the front-end on which the compiler depends is called *runtime*. It provides compiler-specific features in the (otherwise compiler independent) back-end. For example, it handles the instantiation of the boxes, since they are compiler-specific (e.g., the join operation used in SPARQL is slightly different from joins used in SQL). SPARQL runtime provides boxes that represent operations used in SPARQL evaluation. Examples of such boxes are *scan*, *join*, *union*, *filter* box etc. Some of the operations have different implementations. For example, scan box is implemented as full-table scan using direct access to the triples table but also as an indexed access to the table. Join boxes use two basic approaches: nested-loops join and merge-join (faster, but requires ordered inputs). Most other boxes use only one implementation.

V. Experiments

We performed a number of experiments to test functionality and performance of the SPARQL query engine. The experiments were performed using the SP^{2}Bench [5] query set, since this benchmark is considered to be standard in the area of semantic processing. The compiler output was visualized to check the correctness of the plans and the whole query engine was benchmarked against a set of test queries on differently sized data sets to determine. We also performed the same tests on the Sesame [13] SPARQL engine, so we can compare these two SPARQL query engines.

A. Set-up

Experiments were performed on a server running Redhat 6.0 Linux. Server configuration is 2x Intel Xeon E5310, 1,60Ghz (L1: 32kB+32kB L2: 2x4MB shared) and 8GB RAM. It was dedicated specially to the testing; therefore no other CPU or memory services were running on the server. As the benchmark framework (queries and data) we chose the SP^{2}Bench [5] framework that is targeted on testing SPARQL engines and provides a set of queries, and a data generator that creates DBLP-like publication database.

SPARQL front-end and Bobox are implemented in C++. Document data were stored in-memory. We also tested Sesame v2.0 engine using its in-memory data store. We report the total elapsed time that was measured by a timer.

For all scenarios, we carried out multiple runs over documents containing 10k, 50k, 250k, 1M, and 5M triples and we provide the average times. Each test run was also limited to 30 minutes (the same timeout as in the original SP^{2}Bench paper). All data were stored in-memory, as our primary interest is to compare the basic performance of the approaches rather than caching etc. The expected number of the results for each scenario can be found in Table I.

B. Discussion of the Benchmarks Results

The query execution times are shown in Figure 4. The y-axes are shown in logarithmic scale and individual plots scale differently. In following paragraphs, we discuss some of the queries and their results.

Q2 implements a bushy graph pattern and the size of the result grows with the size of the queried data. We can see that Bobox scales well, even though it creates execution plans shaped as a left-deep tree. This is due to the parallel stream processing of fast merge joins.

The variants of Q3 (labeled *a* to *c*) test FILTER expression with varying selectivity. We present only the results of Q3c as the results for Q3a and Q3b are similar. The performance of Bobox is negatively affected by the simple statistics implementation used to estimate the selectivity of the filter.

Q4 (Figure 5) contains a comparably long graph chain, i.e., variables ?name1 and ?name2 are linked through articles that (different) authors have published in the same journal. Bobox embeds the FILTER expression into this computation, instead of evaluating the outer pattern block and applying the FILTER afterwards and propagates the DISTINCT modifier closer to the leaves of the plan in order to reduce the size of the intermediate results. This provides better performance than Sesame.

Queries Q5a (Figure 5) and Q5b test implicit join encoded in FILTER condition (Q5a) and explicit (Q5b) variant of joins. While on explicit join (Q5b) both engines performs similarly, on implicit join (Q5a) Bobox outperforms Sesame since it is able to compute also documents with 250k and 1M triples before the 30 minute limit is reached. This is achieved by creating bushy execution plan (thanks to

 ISBN: 978-1-61208-175-5

	Q1	Q2	Q3a	Q3b	Q3c	Q4	Q5a/b	Q6	Q7	Q8	Q9	Q10	Q11
10k	1	147	846	9	0	23.2k	155	229	0	184	4	166	10
50k	1	965	3.6k	25	0	104.7k	1.1k	1.8k	2	264	4	307	10
250k	1	6.2k	15.9k	127	0	542.8k	6.9k	12.1k	62	332	4	452	10
1M	1	32.8k	52.7k	379	0	2.6M	35.2k	62.8k	292	400	4	572	10
5M	1	248.7k	192.4k	1.3k	0	18.4M	210.7k	417.6k	1.2k	493	4	656	10

Table I
QUERY RESULT SIZES ON DOCUMENTS UP TO 5M TRIPLES.

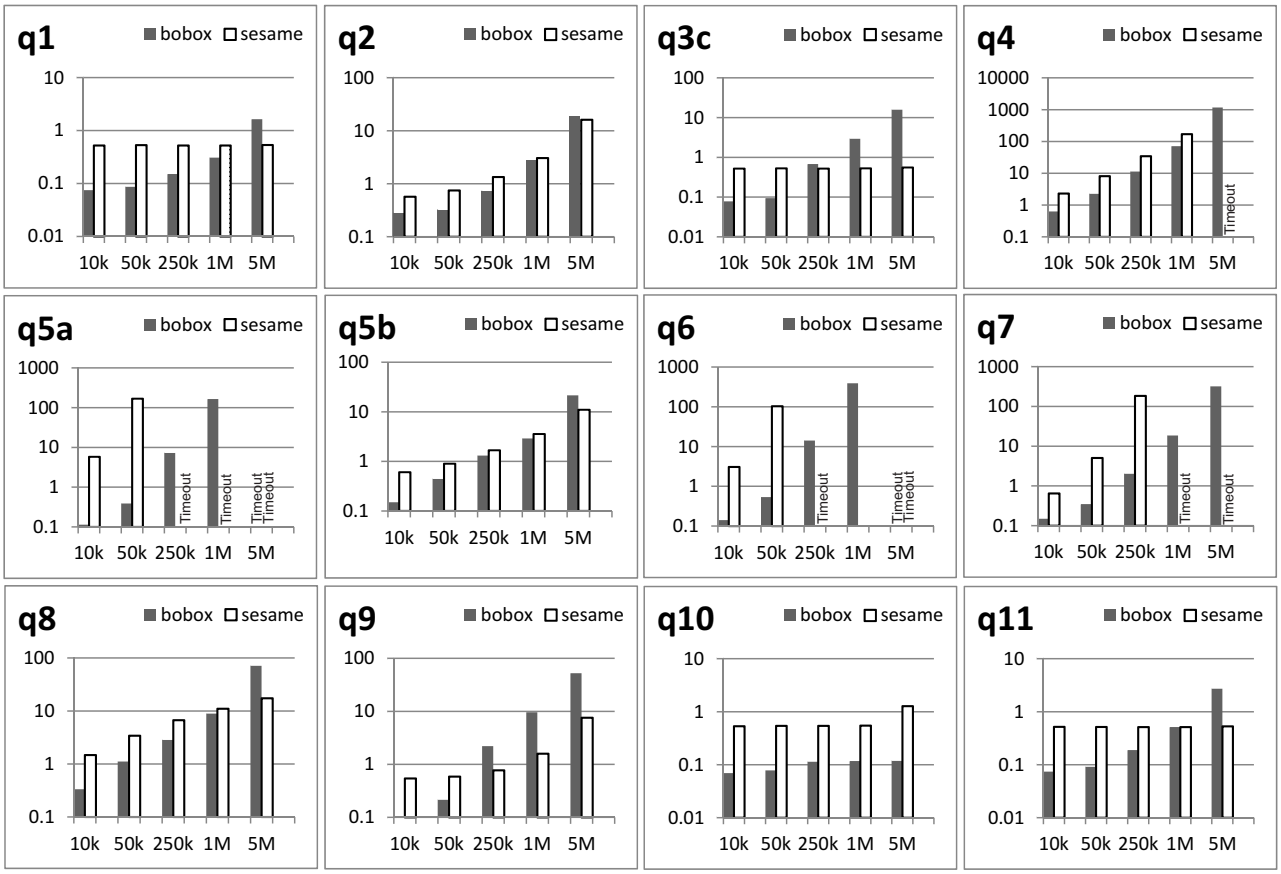


Figure 4. Results (time in seconds) for 10k, 50k, 250k, 1M, and 5M triples.

the rule of minimizing the number Carthesian products) whose execution scales well when executed in parallel. Also, incorporating FILTER operation into the final join, which would otherwise create a Carthesian product, reduces intermediate data size and speeds up query evaluation.

Queries Q6, Q7 and Q8 enable us to create bushy trees, so their computation is well handled in parallel. As a result of this, Bobox outperforms Sesame in Q6 and Q7, being able to compute larger documents until the query times out. The authors of the SP^{2}Bench suggest reusing graph patterns in description of the queries Q6, Q7 and Q8 [5]. However, this is problematical in Bobox. Bobox processing is driven by the availability of the data on inputs but it also incorporates methods to prevent the input buffers from being overfilled. Pattern reusing can result in the same data being sent along two different paths in the pipeline running at a different speed. Such paths may then converge in a join operation. When the faster path overfills the input buffer of the join box, the computation of all boxes on paths leading to the box is suspended. As a result, data for the slower path will never be produced and will not reach the join box, which results in a deadlock. We intend to examine the possibility of introducing a buffer box, which will be able to store and provide data on request. This way, the Bobox SPARQL implementation will be able to reuse graph patterns.

Overall, results of the benchmarks indicate good potential of the Bobox framework when used as an RDF query engine. It is often comparable to the Sesame framework and in

 ISBN: 978-1-61208-175-5

```
SELECT DISTINCT ?name1 ?name2                                   Q4
WHERE { ?article1 rdf:type bench:Article.
        ?article2 rdf:type bench:Article.
        ?article1 dc:creator ?author1.
        ?author1 foaf:name ?name1.
        ?article2 dc:creator ?author2.
        ?author2 foaf:name ?name2.
        ?article1 swrc:journal ?journal.
        ?article2 swrc:journal ?journal
        FILTER (?name1<?name2) }

SELECT DISTINCT ?person ?name                                   Q5a
WHERE { ?article rdf:type bench:Article.
        ?article dc:creator ?person.
        ?inproc rdf:type bench:Inproceedings.
        ?inproc dc:creator ?person2.
        ?person foaf:name ?name.
        ?person2 foaf:name ?name2
        FILTER(?name=?name2) }
```

Figure 5. Examples of the benchmark queries.

some benchmarks it was able to process larger documents and/or outperform it. However, there are still some scenarios, in which Sesame performs better and we are working to improve our implementation to handle these cases better.

VI. Conclusion and Future Work

In the paper, we presented a parallel SPARQL processing engine that was built using the Bobox parallelization framework. Our main focus was on efficient query processing: parsing, optimization, transformation and parallel execution. To store the data, we implemented a simple in-memory triple store. To test performance of our pilot implementation, we performed multiple experiments. We have chosen an established framework for RDF data processing Sesame as the reference system.

The results seem very promising; using SP^2Bench queries we have identified that on simple queries we are in most cases comparable to Sesame. For more complicated queries like Q4, Q5, Q6 or Q7 we are able to process larger documents than Sesame. These queries let us produce richer execution plans; we are able to incorporate FILTER expressions into computation better and together with the use of fast merge joins their execution in parallel gives better performance. However, we also detected some bottle-necks. Our heuristics sometimes result in long chains but streamed processing and fast merge joins minimize this disadvantage. Also, some proposed methods, such as graph pattern reuse are not applicable in our system. During the benchmarking we also discovered some new ideas of how to increase performance of generated plans by query modification and also better use of statistics. We are, therefore, convinced that there is still space for optimization in RDF processing.

We proved that the parallel approach to RDF data processing using the Bobox framework has potential to provide better performance than current serial engines.

Acknowledgments

The authors would like to thank the GAUK project no. 28910, 277911 and SVV-2010-261312, and GACR project no. 202/10/0761, which supported this paper.

References

[1] E. Prud'hommeaux and A. Seaborne, "SPARQL Query Language for RDF," W3C Recommendation, 2008.

[2] D. Bednarek, J. Dokulil, and J. Yaghob, "Bobox: Parallelization framework for data processing," 2011, iET Software - submitted for publishing.

[3] D. Bednarek, J. Dokulil, J. Yaghob, and F. Zavoral, "The bobox project - parallelization framework and server for data processing," Charles University in Prague, Technical Report 2011/1, 2011.

[4] M. Schmidt, T. Hornung, N. Küchlin, G. Lausen, and C. Pinkel, "An Experimental Comparison of RDF Data Management Approaches in a SPARQL Benchmark Scenario," in *ISWC*, Karlsruhe, 2008, pp. 82–97.

[5] M. Schmidt, T. Hornung, G. Lausen, and C. Pinkel, "Sp2bench: A sparql performance benchmark," *CoRR*, vol. abs/0806.4627, 2008.

[6] *OpenMP Application Program Interface, Version 3.0*, OpenMP Architecture Review Board, May 2008, http://www.openmp.org/mp-documents/spec30.pdf, retrieved 9/2011.

[7] A. Kukanov and M. J. Voss, "The foundations for scalable multi-core software in Intel Threading Building Blocks," *Intel Technology Journal*, vol. 11, no. 04, pp. 309–322, November 2007.

[8] *MPI: A Message-Passing Interface Standard, Version 2.2*, Message Passing Interface Forum, September 2009, http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf, retrieved 9/2011.

[9] Z. Falt and J. Yaghob, "Task scheduling in data stream processing," in *Proceedings of the Dateso 2011 Workshop*, 2011, pp. 85–96.

[10] H. Pirahesh, J. M. Hellerstein, and W. Hasan, "Extensible/rule based query rewrite optimization in starburst," *SIGMOD Rec.*, vol. 21, pp. 39–48, June 1992.

[11] O. Hartig and R. Heese, "The SPARQL Query Graph Model for query optimization," in *The Semantic Web: Research and Applications*, ser. Lecture Notes in Computer Science, E. Franconi, M. Kifer, and W. May, Eds. Springer Berlin / Heidelberg, 2007, vol. 4519, pp. 564–578.

[12] M. Cermak, J. Dokulil, and F. Zavoral, "Sparql compiler for bobox," *Fourth International Conference on Advances in Semantic Processing*, pp. 100–105, 2010.

[13] J. Broekstra, A. Kampman, and F. v. Harmelen, "Sesame: A generic architecture for storing and querying RDF and RDF schema," in *ISWC '02: Proceedings of the First International Semantic Web Conference on The Semantic Web*. London, UK: Springer-Verlag, 2002, pp. 54–68.

 ISBN: 978-1-61208-175-5

An Ontology-based Framework for Enriching Event-log Data

Thanh Tran Thi Kim, Hannes Werthner
e-commerce group
Institute of Software Technology and Interactive Systems, Vienna, Austria
Email: kimthanh@ec.tuwien.ac.at, werthner@ec.tuwien.ac.at

***Abstract*—Using process-aware information systems in enterprises is becoming popular in the business environment. The systems have the capability to generate event log data that capture information about what is practically happening within enterprises. Event log data is used for process mining to extract the hidden knowledge which can assist the manager in business process management. However, the knowledge hidden in event logs would be more useful if the event logs are enriched by relevant external data sources. In this paper, we propose an approach to enrich event logs with external data sources by using ontology based data integration. We use database-to-ontology mapping techniques to integrate data sources and use semantic reasoning techniques for inferring the knowledge hidden in the data sources. A framework for the approach, illustrating examples for the implementation and expected results are presented in this paper.**

Keywords-process-aware information systems; data integration; process mining.

I. INTRODUCTION

Process-aware Information Systems (PAISs) are increasingly used by many enterprises in the modern business environment. A PAIS is defined as a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models [1]. Moreover, the system has the capability to generate event log files, which record the information of real executions within enterprises. The knowledge hidden in the event logs is extracted by process mining techniques and used for model construction and analysis [2]. In particular, process mining application includes features of three categories: model construction, statistical performance analysis and knowledge discovery. Model construction refers to the dynamic building of business process based on the information contained in event logs. Statistical performance analysis aims to extract predefined statistical measures. Knowledge discovery is the incorporation of event log data with other data sources to search for hidden patterns and relationships [3]. Several studies have been carried out to show the potential of this incorporation. Most of them use data warehouse techniques for integrating data sources and extracting knowledge from the data sources [3], [4]. However, complexity problems are raised as challenges for this approach [4], [5]. Workflow executions may generate different kinds of facts about workflow activities, resources, and instances. Because of the multiple, related types of facts, the approach may be faced with semantic problems. Particularly, the presence of these kinds of facts needs to ensure semantic correctness to avoid information loss [5].

To avoid the problems of the data warehouse approach, we propose the framework for integrating event logs with other data sources based on the TOVE ontology [6], [7]. TOVE (TOronto Virtual Enterprise) is an integrated ontology for supporting enterprise modeling which contains concepts related to business models, such as activity, organization agent, cost, resources, etc. Event logs are exported by PAISs to record the operations of business processes in companies, such as the information about who performs which activities at what time. The approach is raised by the question how to enrich event log data and what knowledge could be gained from the enrichment. Merging data in event logs with other data sources are mentioned in [3] as a potential approach for knowledge discovery in process mining. The benefit of the approach could be seen in the enriched event logs which is extended with relevant information by linking to ontologies. Therefore, the knowledge extracted from event logs is collected not only from the event logs but also from others company related data sources, which are related and linked to them. For instance, cost data is not included in event logs but can be inferred by reasoning from the cost ontology in TOVE. Therefore, the results of process mining in can be opened to new perspectives, e.g., cost perspective.

In general, our approach contains two main parts: ontology based data integration and knowledge discovery. Ontologies are very useful in knowledge sharing and integration as well as knowledge research and extraction [8]. In our study, we use TOVE ontology as a conceptual framework for integrating data sources. In particular, event log data and organizational data are migrated to TOVE ontology as instances. Hence, TOVE becomes a knowledge base and can be used for knowledge discovery. As a result, competency questions related to business process management can be answered by querying the axioms constructed in TOVE.

The remainder of the paper is structured as follows: Section 2 introduces the various data sources and the TOVE ontology which are the main objects of the integration. Section 3 presents the framework for mapping event logs and other data sources to the TOVE ontology. Section 4 illustrates the querying axioms for answering questions related to business process management and the expected results.

 ISBN: 978-1-61208-175-5

Section 5 presents the related work, including knowledge discovery in process mining, semantic process mining and TOVE ontology. Finally, Section 6 concludes the paper.

II. THE TOVE ONTOLOGY AND VARIOUS DATA SOURCES

A. The TOVE ontology

TOVE is an integrated set of ontologies for supporting enterprise modeling [9]. The development of the TOVE ontology is driven by the specification of tasks that arise from enterprise engineering within the TOVE project [7]. The goal of enterprise engineering is to formalize the knowledge required for business process reengineering and create an environment that facilitates the application of this knowledge to a particular company. The ontology consists of a set of generic core ontologies, including an activity ontology, resource ontology, organization ontology, product ontology. It also includes a set of extensions to these generic ontologies to cover concepts such as cost and quality.

The primary component of the ontology is its terminology for classes of processes and relations of processes and resources, along with definitions of these classes and relations. Within TOVE, the activity ontology plays an important role and relates to most of axioms [9], [10]. In TOVE, activities are defined as the basic entities that specify a transformation in the world. An activity in TOVE is accompanied with its corresponding states which defines what has to be true in the world in order for the activity to be performed. Moreover, an activity is performed by an organization agent with a particular amount of resources. Based on the relations between activity, organization, resource ontologies, most of questions related to enterprise management are satisfied by querying the axioms built in the ontology. Another prominent part of TOVE is the cost ontology. Costs are related to consuming resources and time when performing activities. Figure 1 shows a set of generic core ontologies in TOVE.

The TOVE ontology presents a mature framework whereas event log data have a simple data structure. Event logs contain information about activites, originators who perform the activities, the process instances which the activites belong to, and the timestamp when the activities occur. Opposite with the simplicity of event log data, TOVE contains many concepts, as shown in Figure 1 and most of the concepts of TOVE are not related directly to event log data elements. Therefore, we use a part of TOVE which are simplified to be suitable with the event log data. For example, the activity ontology in TOVE has relations with the product requirement constraints concept. However, we bypass the product requirement constraints concept because the data of the product requirement constraints do not exist in event log data.

In our approach, we select the activity ontology, organization ontology, resource ontology and cost ontology. In addition, we add a new concept to TOVE (i.e., process concept) and modify some properties of concepts in TOVE to correspond with the properties of event log data and organization database. The knowledge derived from TOVE will be used to enhance process models as results of the process mining.

B. Various data sources

The different data sources in our project are event log data and organization databases. We assume that in companies which are using information systems to support business management, event log data can be received from a PAIS and organization databases obviously exist in a particular database system. The details of event log data and organization database are described as follows.

PAISs produce event log files to record the operation of business processes. Depending on the particular PAISs in use, event log data may contain various types of information in different formats. Generally, an event log data record is consisting of an activity (task name), originator, timestamp, event type and case identification elements [2]. The activity element indicates the name of the activity or the task which is operated. Originator implies entities who initiate or perform the activity. Timestamp is the point of time when the activity happened. Event type denotes the state of the activity (e.g., the start or completion or postpone of the activity). And case identification is a unique number that identifies a specific process instance to which the activity belongs. Although the contents of a log data record may vary, event logs need to contain at least activity and case identification elements.

As the example of Table I shows, *activity A* was performed by Mark at the time *17-05-2008:16:09*; the activity was in the *start* state and belongs to the *case 1*. *Case 1* includes a number of activities, such as *activity A* and *activity B*. All the activities are ordered by their respective timestamp.

Table I
EXAMPLE OF AN EVENT LOG

case id	activity id	originator	timestamp	event type
case 1	activity A	Mark	17-05-2008:16:09	start
case 2	activity B	Chris	18-05-2008:09:12	start
case 1	activity C	Tom	18-05-2008:10:06	complete
case 3	activity B	Mary	18-05-2008:15:02	start
...	...	...	...	...

In terms of semantics, a log file refers to a set of process instances (i.e., cases). Each process instance includes a number of events happened within the process. An event occurs when an activity is operated by an originator at a certain point of time (i.e., timestamp). Each event has an event type representing the status of the event when it is performed, e.g., start or complete. Hence, one can observe that TOVE ontological concepts for enterprise operation are considerably similar to the concepts appearing in event logs.

Considering the data fields in event logs, there is the originator element which contains information of employees

 ISBN: 978-1-61208-175-5

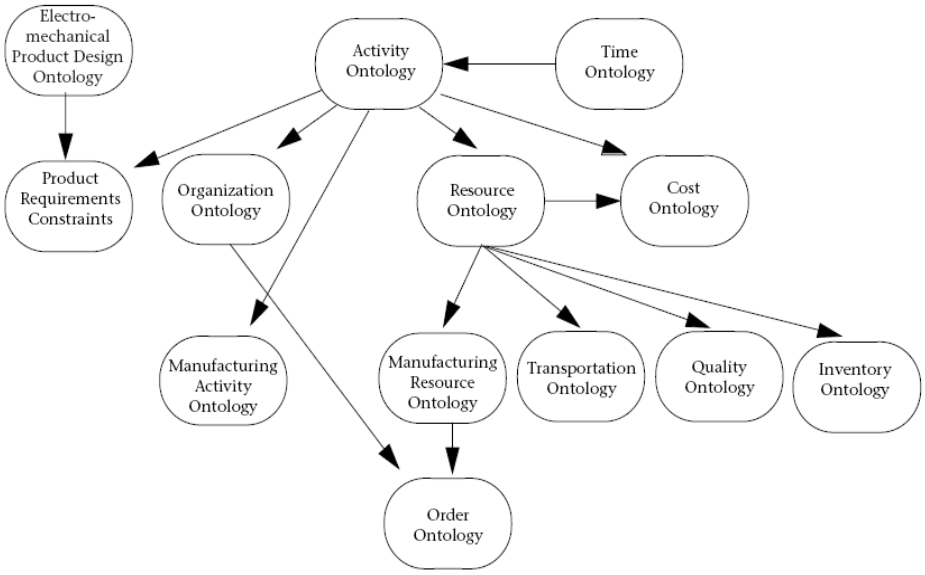


Figure 1. TOVE Ontology [11]

who perform activities. For business management, it is obvious to store the data of employees in a database, i.e., an organizational database with data schema as presented in Figure 2.

The important tables in the database are *employee* and *activity* which are related directly to *originator* and *activity* respectively in event log data. Based on the properties in these table, the information of *originator* and *activity* in event log could be extended. For instance, an originator has information about address, experience year or the labour cost, etc.

III. Framework for integrating event logs and other data sources based on the TOVE ontology

There are two main functionality blocks in the framework: mapping and knowledge discovery. In this context, mapping refers to the adding of instances into the TOVE ontology from data sources. The derived result of the mapping is the TOVE ontology with instances which is regarded as a knowledge base. Knowledge discovery is performed by querying axioms in the knowledge base. Figure 3 represents briefly the framework of the ontology based integration in our approach.

We have two types of data sources, event log data and organizational database. As mentioned in Section 2, we suppose event log data contains information about activities, originators, timestamps, and cases identifications. Organizational database contains the information support for enterprise management dealing with cost accounting, human resources management or resources. The mapping

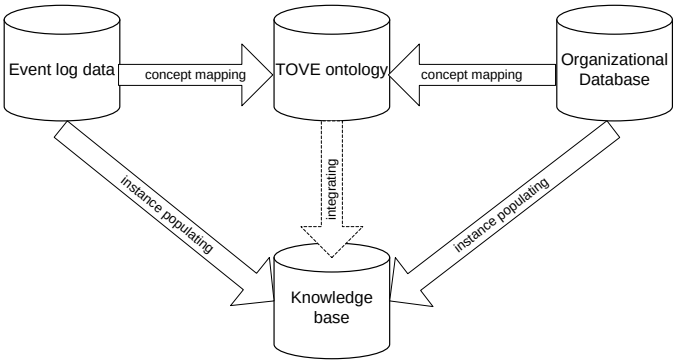


Figure 3. Mapping event log data and extra data sources to the TOVE ontology.

from event log data to TOVE ontology is considered as the migration instances from data fields (i.e., activity, originator and case) to concepts (i.e., activity, organization-agent and process) respectively. Particularly, the values of the name properties in TOVE ontology is filled by the values of the data fields in the event logs. The values of the rest of the properties in TOVE are filled by the values of data fields in the organizational database. The mapping is referred to database-to-ontology mapping whereby a database and an ontology are semantically related at a conceptual level [12], [13]. In our approach, we assume the concept of originator in event logs is similar to an organization agent in TOVE. Likewise, event and timestamp correspond to activity and timestamp, respectively. Therefore, the integration based on

 ISBN: 978-1-61208-175-5

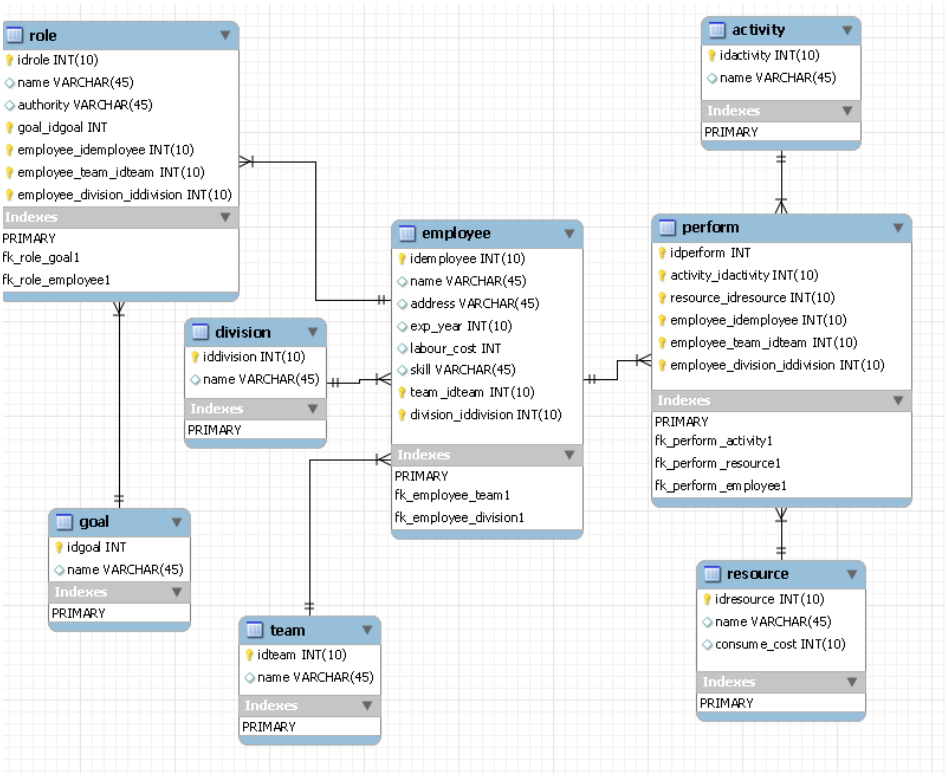


Figure 2. Organizational Database

the TOVE ontology is feasible.

Using reasoning techniques over the ontologies can discover knowledge hidden in the data sources. The reasoning is done by querying the axioms in the TOVE ontology. Note that there are a huge number of axioms in TOVE which support for answering the question related to enterprise management and modeling [9], [10], [14]. Thus, the reasoning may be valuable for knowledge discovery. As a result, combining the semantic reasoning and process mining techniques for discovering knowledge in the enriched event log data represents a sound approach for semantic process mining.

To implement the framework, we use Java [15] as a foundation to combine several techniques. In particular, the event log files are stored in XML format and the organizational database is managed by MySQL [16] . The TOVE ontology and the knowledge base are encoded and stored in WSML format [17]. Besides, several java packages are utilized for data integration (e.g., javax.xml.xpath, java.sql, etc.) and knowledge extraction (e.g., wsmo4j). Within this paper, we introduce a part of knowledge base and the expected results of the knowledge extraction in Section 4.

IV. QUERYING AXIOMS FOR ANSWERING QUESTIONS RELATED TO BUSINESS PROCESS MANAGEMENT

As a result of the ontology-based data integration process, we obtain an knowledge base containing event log data and organizational data. In this section, we illustrate an example about querying axioms for answering questions related to costs of business processes. Figure 4 shows a part of the knowledge base as a diagram of concepts with their properties.

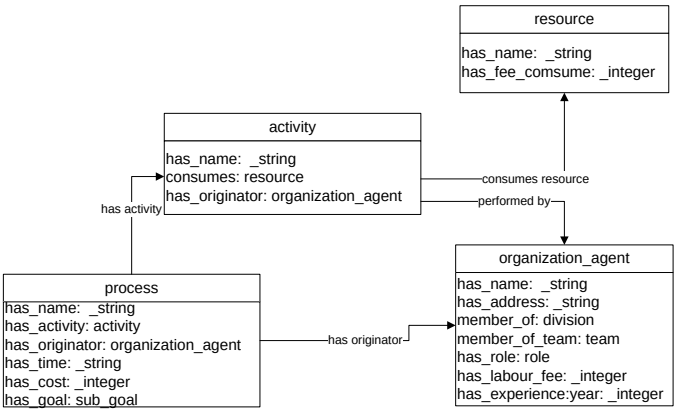


Figure 4. A part of the knowledge base

There are four concepts *resource, activity, organization-agent and process*. They are related by the relationships *consumes resource, performed by, has activity, has originator*. Considering the concept *process*, it is an additional concept which is added to TOVE to use information about process

instances in event log data. Based on this concept, questions related to processes can be answered.

Deriving costs of business processes is currently not possible with process mining. In our approach, an interesting question that can be answered is "How much does a process cost"?. We use the WSML toolkit for building the ontology and testing axioms as shown in Figure 5.

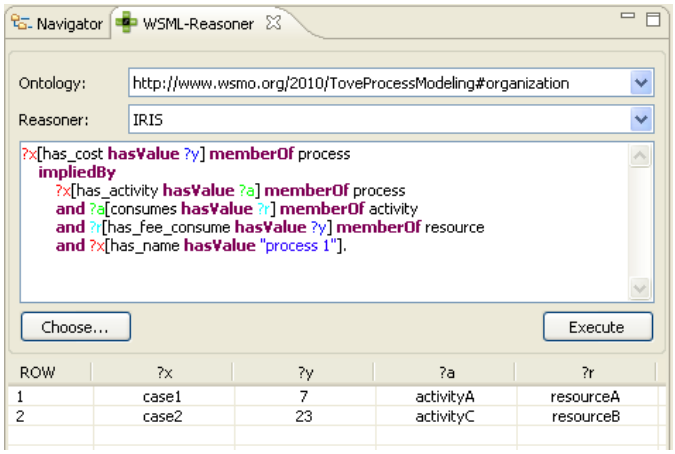


Figure 5. Reasoning with WSML toolkit

Figure 5 displays an axiom of the knowledge base in WSML format for costing a process. WSML utilizes logical expression syntax for the specification of axioms, in other words, rules are defined as logical expression in WSML. In the example, the rule "how much does a process (e.g., *process 1*) cost?" is demonstrated. In detail, the *process 1* is defined by two instances of the concept *process* (i.e., *case1* and *case2*). The *process 1* has two activities, *activityA* and *activityC*. Each activity consumes resources which have particular costs associated. In this case, the *resourceA* has cost 7 and *resourceB* has cost 23 which are values of the property *has_fee_consume* . Therefore, the cost of the *process 1* is inferred from the cost of *resourceA* and *resourceB* which are used by *activityA* and *activityC* respectively.

Moreover, based on the constraints between the concepts shown in Figure 4, various kinds of questions can be answered, such as:

- How much does the consumption of resources cost for performing *activity A* in *process 1*?
- Which resources are consumed in *process 1*?
- How much does it cost for performing *process 1*?

V. Related Work

Knowledge discovery in process mining by incorporating event logs with other data sources is mentioned in [3], [4], [5]. Most of the authors use a data warehouse approach for integrating and extracting knowledge from the data sources. It provides a platform for mining unknown and valuable patterns and relationships. Some of the significant techniques in this area, such as OLAP (online analytical processing), traditional database queries, data mining, and etc., are used in this field. OLAP technology enables data warehouses to be used effectively for online analysis, providing rapid responses to interactive complex analytical queries [3]. On the other hand, traditional database queries can answer simple questions. In contrast, data mining with specific algorithms can identify discernible patterns and trends in data, and it can support prediction and decision making. The merging of data from event logs with other data sources are carried out within several studies [3], [4], [18].

Process mining aims to discover what really happened in the enterprise systems based on event logs recorded by PAISs. Depending on the kind of information contained in event logs, the process mining is separated into three perspectives, i.e., process perspective, organizational perspective and case perspective which respectively answers the question "How?", "Who?" and "What?" [2]. The results delivered from process mining might be process models, analysis diagrams, or answers for questions involved to business process management. Although some process mining algorithms are borrowed from data mining or others fields, all of them are developed and adapted for the goals of process mining as mentioned above. The significant capability of process mining is to reveal the hidden knowledge in event logs to aid the enterprises to know what is really going on in their systems [2]. To practice process mining, more than 280 plug-ins have been implemented in ProM [19], [20]. Some of process mining techniques have been implemented as tools and applied in the real systems such as health care systems in hospitals or invoice processing systems, and brought out benefits for the enterprises in the domains [2], [21].

To keep improve the achievements gained in process mining, a new approach has been researched which is called semantic process mining and carried out within the SUPER project [22]. Basically, the methodology is to connect elements in event logs with adequate concepts in ontologies and cooperate the process mining and semantic techniques to deliver on expected results. With this approach, process mining has been raised from the syntactic level to the concept level in which it is more effective and useful for business analysts as well as normal users [23]. Compared with our approach, in the SUPER approach event logs are also enriched by connecting with concepts in ontologies. However, the difference is that the knowledge discovery in the SUPER approach is done by enriching event logs, whereas in our approach it is performed in the TOVE ontology (i.e., the knowledge base). Moreover, with the ontology based integration, the enrichment can be done with different data sources.

VI. Conclusion

This paper proposed a framework for integrating event logs with other data sources and mapping them to on-

tologies and afterwards using these results in semantic process mining. The mapping is termed as database-to-ontology mapping and supported by several existing tools. For this purpose, we use the TOVE ontology, which in our case is populated with instances extracted from different data sources. The integration enriches the event logs with extra information from the other data sources. It serves for answering questions (by reasoning) relating event logs with organizational data. This framework is already implemented and currently evaluated.

REFERENCES

[1] M. Dumas, W. M. van der Aalst, and A. H. ter Hofstede, *Process-aware information systems: bridging people and software through process technology*. New York, NY, USA: John Wiley & Sons, Inc., 2005.

[2] W. M. P. van der Aalst, H. A. Reijers, A. J. M. M. Weijters, B. F. van Dongen, A. K. A. de Medeiros, M. Song, and H. M. W. E. Verbeek, "Business process mining: An industrial application," *Inf. Syst.*, vol. 32, no. 5, pp. 713–732, 2007.

[3] J. E. Ingvaldsen and J. A. Gulla, "Model-based business process mining," *IS Management*, vol. 23, no. 1, pp. 19–31, 2006.

[4] M. zur Mühlen, "Process-driven management information systems - combining data warehouses and workflow technology," in *Fourth International Conference on Electronic Commerce Research*, B. Gavish, Ed., 2001, pp. 550–566, dallas.

[5] A. Bonifati, F. Casati, U. Dayal, and M.-C. Shan, "Warehousing workflow data: Challenges and opportunities," in *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 649–652.

[6] M. S. Fox, M. Barbuceanu, and M. Grüningerr, *An Organisation Ontology for Enterprise Modelling: Preliminary Concepts for Linking Structure and Behaviour*, 1995, pp. 123–134.

[7] E. I. Laboratory, "Tove ontology project," http://www.eil.utoronto.ca/enterprise-modelling/tove/, retrieved: August, 2011.

[8] M. Hepp, "Ontologies: State of the art, business potential, and grand challenges," in *Ontology Management, Semantic Web, Semantic Web Services, and Business Applications*. Springer, 2008, pp. 3–22.

[9] M. Grüninger, K. Atefi, and M. S. Fox, "Ontologies to support process integration in enterprise engineering," *Comput. Math. Organ. Theory*, vol. 6, no. 4, pp. 381–394, 2000.

[10] M. Grüninger and M. S. Fox, "An activity ontology for enterprise modelling," in *Submitted to: Workshop on Enabling Technologies - Infrastructures for Collaborative Enterprises*, West Virginia University, 1994.

[11] M. S. Fox and M. Grüninger, "Enterprise modeling," *AI Magazine*, vol. 19, no. 3, pp. 109–121, 1998.

[12] N. C. Raji Ghawi, "Database-to-ontology mapping generation for semantic interoperability," in *Third International Workshop on Database Interoperability*, 2007.

[13] J. Barrasa, s. Corcho, and A. Gómez-pérez, "R2o, an extensible and semantically based database-to-ontology mapping language," in *in In Proceedings of the 2nd Workshop on Semantic Web and Databases(SWDB2004*. Springer, 2004, pp. 1069–1070.

[14] D. Tham, M. S. Fox, and M. Grüninger, "A cost ontology for enterprise modelling," in *Proceedings of third Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Morgantown,WV*, 1994, pp. 111–117.

[15] Oracle, "Oracle technology network for java developers," http://www.oracle.com/technetwork/java/index.html, retrieved: August, 2011.

[16] "Mysql:: The world's most popular open source database," http://www.mysql.com, retrieved: August, 2011.

[17] S. T. I. S. I. ESSI WSML working group, "Web service modeling language wsml," http://www.wsmo.org/wsml/wsml-syntax#l, retrieved: August, 2011.

[18] D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal, and M.-C. Shan, "Business process intelligence," *Comput. Ind.*, vol. 53, pp. 321–343, April 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=982250.982256

[19] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst, "The proM framework: A new era in process mining tool support." in *Lecture Notes in Computer Science: Applications and Theory of Petri Nets 2005: 26th International Conference, ICATPN 2005, Miami, USA, June 20-25, 2005. / Gianfranco Ciardo, Philippe Darondeau (Eds.)*, vol. 3536. Springer Verlag, Jun. 2005, pp. 444–454.

[20] E. T. U. Process Mining Group, "Prom - the leading process mining toolkit," http://prom.win.tue.nl/tools/prom/, 2009, retrieved: August, 2011.

[21] R. S. Mans, H. Schonenberg, M. Song, W. M. P. van der Aalst, and P. J. M. Bakker, "Application of process mining in healthcare - a case study in a dutch hospital," in *BIOSTEC (Selected Papers)*, 2008, pp. 425–438.

[22] SUPER, "Super integrated project," http://www.ip-super.org/content/view/71/73/, retrieved: August, 2011.

[23] A. K. Alves De Medeiros, W. Van Der Aalst, and C. Pedrinaci, "Semantic process mining tools: Core building blocks," in *16th European Conference on Information Systems*, W. Golden, T. Acton, K. Conboy, H. van der Heijden, and V. K. Tuunainen, Eds., Galway, Ireland, 2008, pp. 1953–1964.

 ISBN: 978-1-61208-175-5

An Experiment on Semantic Emotional Evaluation of Chats

Concepción Bueno, Juan Antonio Rojo, Pilar Rodriguez
Escuela Politécnica Superior, Universidad Autónoma de Madrid,
Francisco Tomás y Valiente, 11
28049 Madrid, Spain
e-mail: diazmunio@hotmail.com, joan.rojo.83@gmail.com, pilar.rodriguez@uam.es

***Abstract*—Electronic conversations always contain an emotional charge. Being able to evaluate such emotional charge is an interesting challenge, and valuable conclusions can be obtained if that process is performed automatically. In this paper, we present a Semantic Emotional Evaluator for Chats, named Chat-SEE, that has been used for evaluating the emotions in a chat conversation. The results obtained are quite promising.**

Keywords- semantic emotional evaluation; cooperative work; chat semantic

I. MOTIVATION

Electronic conversations, as well as any other kinds of conversations, always contain an emotional charge. Being able to evaluate such emotional charge is an interesting challenge, and valuable conclusions can be obtained if that process is performed automatically.

For those reasons, we planned to explore the possibility of designing and implementing an emotional evaluator that allows the measurement of the emotional content within a conversation. The emotional evaluation performed allowed us to research about the evolution of the participant emotions through the conversation. In the experiment carried out, three persons were asked to accomplish a cooperative task only making use of a standard online chat. In addition, they did not know who the other participants were.

In that context, our study aimed at proving that emotions can be measured and, also, that they present some relations among each other. Moreover, we aimed at presenting the results obtained in a visual and clear way.

In this paper, we present a Semantic Emotional Evaluator for Chats, named Chat-SEE, that has been used to evaluate the emotions in a chat conversation. The rest of this work is organized as follows. Section II briefly reviews the state-of-the-art on emotional evaluation. Following, Sections III and IV describe the experiment carried out and Chat-SEE evaluation steps, respectively. We present the results in Section V and the conclusions and future work in Section VI.

II. RELATED WORK

Nowadays, emotional analysis is an outstanding research area that starts offering very interesting results. There are different studies, systems and applications available, which deal with emotion evaluation from diverse points of view. Some of them are based on voice spectrum and stress [1][2] or on gesture and expression analysis [3][4][5], while others try to conclude about the emotion charge of texts. Within this last group, some works have centered in analyzing individual short texts [6][7][8], while others have been applied to cooperative texts involving several users [9][10][11]. Though sharing similar goals, their approaches and final results are different.

For instance, [6] presents an approach to emotion analysis of new headlines. It proposes and evaluates several methods to identify an emotion in text. The emotions gathered are joy, anger, disgust, fear, sadness and surprise, which are used to classify the headlines accordingly. Also focused on headlines, the Headline Analyzer online application [7] aims at measuring the impact of short texts on potential readers, the so called emotional marketing value. In [7], the dimensions taken into account are: intellectuality, empathy and spirituality. Another market oriented application can be found in [8]: SAS Sentiment Analysis. That application analyzes digital content in order to understand customers' opinions. Positive and negative sentiments are inferred.

Regarding works that have centered on collaborative texts, [9] presents a study performed at HP Labs that demonstrates how important is to extract the emotions automatically from text in social media, and how it can be useful to forecast the impact of some topic. In particular they use tweets related to a movie to forecast box office revenues for movies. The emotions extracted from tweets are positive, negative and neutral. Once they extract the emotions from tweets and, applying different formulas, they obtain the positive or negative impact of a movie and, consequently, the higher or lower box offices revenues.

Also, a quite interesting work is presented in [10]. In that study they present a system to extract sentiment from text. It uses an annotated dictionary where a measurement of polarity, strength, intensification and negation are assigned to words. Different dictionaries are used with different results; it demonstrated the vital importance of the dictionary used. It is a content independent based system that has performed well on blog postings and video games reviews without any training process.

Finally, other interesting system is Text Tone [11]. Text Tone allows users to tag emotions in the text introduced in online textual communication, so people can easily understand the meaning of a conversation. It is useful when users try to express an emotion that can be ambiguous or to

 ISBN: 978-1-61208-175-5

emphasize certain emotion. However, Text Tone does not analyze the text introduced by the user. There, users decide on their own emotion charge.

III. THE EXPERIMENT

In this work, we took Gmail conversation chats as starting point. Each chat took place among exactly three people, being all of them students enrolled in the Master on Computer and Telecommunication Engineering at the Universidad Autónoma de Madrid. There were 6 groups, that is, 18 persons involved. Spanish was the language used.

Each group was asked to carry out a collaborative task during two hours. The activity consisted in trying to reconstruct a previously fragmented play script. There were some basic rules: they were not allowed to identify themselves and they did not have to delete anything they do. With this intention, each member of a group was provided with an e-mail address, its password and the e-mail address of his/her partners. Each team member had in the inbox of his/her e-mail a document with some characters of the play and some utterances. The whole play consisted of four characters and forty two utterances. Each group was required to give a joint solution to the activity. In order to do that, they had to gather all the information, attribute utterances to the characters, and chronologically arrange them. The process was unsupervised.

It is not surprising that the final chats became something funny and a little bit chaotic. When reading those chats, it seemed that people had had different attitudes when facing the proposed task, enjoying (or not) themselves during the process.

Then, we tried to determine whether the emotions in the conversation could be somehow evaluated. In that sense, we first had to decide which emotions we would focus on. Finally, we decided to make use of the classification ºproposed in [12]. In that work, four basic emotions are identified: joy, anger, fear and sadness. Authors state that those four basic emotions are directly related to the so named "fundamental challenges" such as danger (leading to fear), separation from positive conditions, including inadequate self-efficiency (leading to sadness), frustration of expectancies and registration of inhibitions (leading to anger) or self-efficiency and social acceptance (producing joy).

Though many other classifications of emotions can be found, as in the systems mentioned in Section II, we thought that the abovementioned classification fits perfectly for the experiment. The emotional meaning attributed to joy, anger, fear and sadness in Chat-SEE environment is briefly explained in next section.

IV. CHAT-SEE

We have implemented Chat-SEE in a modular way, and based on three different modules: the dictionary, the tagger and the graph generator.

In addition, conversations are first converted to an XML format, so the rest of the process can be afforded easily. Programming language was Python, making use of the Natural Language Toolkit (NLTK) offered [13] [14].

In the rest of this section, the three modules are described. The examples and graphs presented are taken from a couple of chats, which correspond to groups A and B. Members of group A are identified as Huey, Dewey and Lowie, and members of group B as Kate, Jack and James.

A. Dictionary

Firstly, we created a dictionary based on the words that we had found in the chat to be used in this experiment. At this stage, no preprocessing, stemming or other NLP techniques were used. That decision was taken because of the characteristics of the texts: lots of misspellings and abbreviations.

In that process, not all the words presented in the conversations were tagged. The only words tagged were those that were supposed to have an emotion charge in the chat context.

The chat texts were initially XML formatted, so human judges could easily assign values to the different emotional dimensions chosen. More of a hundred of words were tagged, apart from some commonly used emoticons, what represented about 6-7% of the total words in the chats. In average, the total number of words in the chats was around 1500. For emotion quantification, it was decided to use a range between 0 and 3 (0 minimum and 3 maximum).

Regarding the meaning attributed to joy, anger, fear and sadness in Chat-SEE environment, it slightly differs from the meaning used in [12], being adapted to what a single word can express in terms of emotions. So, "anger" was also supposed to express a kind of criticism, as in the word "no", whose entry is:

<**word token**="no" **joy**="0" **anger**="2" **fear**="1" **sadness**="1" />

Also in that entry, a value of 1 for fear and sadness is attributed.

Other entries are simpler, like "ok", that only seems to express some kind of "approval", which is associated to joy:

<**word token**="ok" **joy**="2" **anger**="0" **fear**="0" **sadness**="0" />

Some cross-checking of the emotion assignment was done in order to detect judge dependencies but most of the assignments were identical, or almost identical.

B. Tagger

The second stage in the emotional evaluator development is the creation of a parser-tagger. The main function of this parser-tagger is to isolate and to emotionally classify each word in an XML file. As was mentioned above, the creation of a structured file makes easier the measurement process for each user intervention. Also, NLTK Python module was used at this stage in order to carry out the process of word extraction and detection. Words are searched in the dictionary and, each utterance emotions are measured and assigned to each user intervention. The global emotions

 ISBN: 978-1-61208-175-5

correspond to the sum of all the word emotions that appeared in the utterance.

Once we had all the emotional scores associated to each utterance, then we create a new XML file with the utterance scores. This file is the input for the last module of Chat-SEE: the Graph Generator. Apart from the emotion information per utterance, such file also includes a time stamp that makes possible to determine when each utterance had taken place.

Following, it is included part of the chat at this stage. It corresponds to the conversation taking place at the 27th minute within group A, among Huey, Dewey and Lowie. As it can be observed, during that minute Huey and Lowie make two contributions, whereas Dewey makes just one. In the text, **j**, **a**, **f** and **s** stand for joy, anger, fear and sadness, respectively.

```
<time id="16:51">
    <user id="Huey">
        <utterance>
          <word j="1" a="2" f="1" s="1" token="ya"/>
          <word j="0" a="1" f="0" s="0" token="veo"/>
        </utterance>
        <utterance>
          <word j="0" a="1" f=0" s="0" token="hay"/>
         </utterance>
    </user>
    <user id="Louie">
        <utterance>
          <word j="0" a="1" f="0" s="0" token="ver"/>
        </utterance>
    </user>
        <user id="Dewey">
        <utterance>
          <word j="2" a="0" f="0" s="0" token="vale"/>
          <word j="0" a="2" f="1" s="1" token="no"/>
          <word j="0" a="2" f="0" s="1" token="pero"/>
          <word j="0" a="1" f="1" s="1"
token="entender"/>
          <word j="0" a="1" f="1" s="1"
token="orden"/>
        </utterance>
    </user>
    <user id="Huey">
        <utterance>
        </utterance>
    </user>
    <user id="Louie">
        <utterance>
        </utterance>
        <utterance>
        </utterance>
</time>
```

As can be seen, during the 27th minute Huey contributed twice to the chat, but only his first contribution had any emotion charge.

C. Graph Generator

Finally, Chat-SEE generates visual representations of the emotional evaluations by making use of standard graph generators, like GNUPLOT.

The Graph Generator of Chat-SEE works as follows. Firstly, it checks about the chat participants, and auxiliary files are generated for any of them, separately. Secondly, aggregation files are created for any of the utterances, by adding the emotion charges corresponding to each word. For Huey's 27th minute, the result is:

```
<time id="16:51">
    <user id="Huey">
        <utterance w="2" j="1" a="3" f="1" s="1">
        </utterance>
        <utterance w="1" j="0" a="1" f="0" s="0">
         </utterance>
    </user>
    <user id="Huey">
        <utterance w="0" j="0" a="0" f="0" s="0">
        </utterance>
    </user>
    </time>
```

In the former text, **w** indicates the number of words with emotion charge in each utterance, while the individual words have been eliminated.

Next, the resulting value assigned to each utterance is aggregated together with the values assigned to the rest of utterances that took place at that very minute. That value is divided among the number of contributions that had taken place at the same time. So, the final emotion media per user and minute are obtained. For Huey's 27th minute, we obtain a emotion media of 0,5 joy, 2 anger, 0,5 fear and 0,5 sadness.

But, in this experiment, analyzing the evolution of the participant emotions was also a challenge. So, another kind of graphs was foreseen. In those graphs, the evolution of the participant emotion would be represented. Those graphs should smooth out the variation intensity of the participant emotions in the period under study.

For generating those smoothed-graphs, the emotion media previously calculated is divided by the number of instants (minutes in this case) since the beginning of the chat. Tables 1, 2 and 3 show all the emotion data for the members of group A (Huey, Dewey and Louie): utterances, contributions, emotion media and smoothed out emotion media at the 27th minute.

TABLE I. HUEY'S EMOTION CHARGE, 27TH MINUTE

		BASIC EMOTION VALUES			
CONTRIBUTION	**UTTERANCE**	**Joy**	**anger**	**fear**	**sadness**
contribution 1	utterance 1	1	3	1	1
	utterance 2	0	1	0	0
contribution 2	utterance 1	0	0	0	0
EMOTIONS		0,5	2	0,5	0,5
SMOOTHED EMOTIONS		0,8	0,57	0,17	0,36

 ISBN: 978-1-61208-175-5

TABLE II. DEWEY'S EMOTION CHARGE, 27TH MINUTE

		BASIC EMOTION VALUES			
CONTRIBUTION	**UTTERANCE**	**Joy**	**anger**	**fear**	**sadness**
contribution 1	utterance 1	2	6	3	4
EMOTIONS		2	6	3	4
SMOOTHED EMOTIONS		1,37	0,94	0,35	0,64

TABLE III. LOUIE'S EMOTION CHARGE, 27TH MINUTE

		BASIC EMOTION VALUES			
CONTRIBUTION	**UTTERANCE**	**Joy**	**anger**	**fear**	**sadness**
contribution 1	utterance 1	0	1	0	0
contribution 2	utterance 1	0	0	0	0
	utterance 2	0	0	0	0
EMOTIONS		0	0,5	0	0
SMOOTHED EMOTIONS		0,29	0,5	0,22	0,35

V. RESULTS

After Chat-SEE execution, three different kinds of graphs are obtained: instant emotion media per participant graph, smoothed out emotion evolution per participant graph and smoothed out chat evolution per emotion graph.

Figure 1 depicts Huey's emotion media during the 70 minutes that the experience lasted. In Figure 1, x-axis corresponds to moments (in minutes) and y-axis corresponds to the instant emotion intensity. In this kind of graphs, it is possible to detect when the emotion peaks took place at a glance. For example, in Figure 1 it is possible to observe that Huey's maximum "joy" happened a little bit after the 40th minute.

Regarding the second kind of graphs, which represent the smoothed out emotion evolution per participant, an example is presented in Figure 2, where x-axis corresponds to moments (in minutes) and y-axis corresponds now to the smoothed out emotion intensity. There, Huey's smoothed out emotion evolution is presented. Firstly, Huey seems to be quite expressive. Moreover, his "joy" line is high, and it surpasses the rest of his emotions. One possible interpretation is that Huey was motivated at accomplishing the proposed task and enjoyed himself while performing it.

Also, Huey "anger" line is not so relevant. It might be because, though he enjoyed himself, he did not take a leadership role.

Finally, Figures 3 to 6 represent the smoothed out emotions of the above mentioned groups, A and B, along the

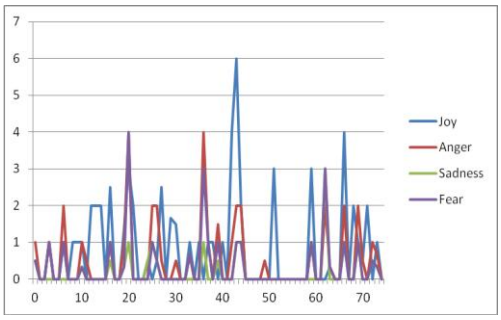


Figure 1. Huey's instant emotion media. A "Joy" peak takes place around the 40th minute of the experiment.

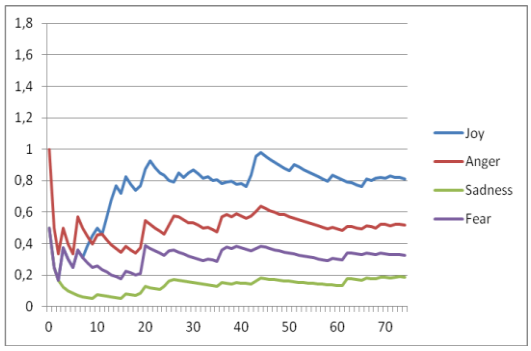


Figure 2. Huey's smoothed out emotion evolution during the experience.

time. Both groups took part in the same experiment, as described in Section III. Those graphs represent the smoothed out chat evolution per emotion graph for both groups.

In those figures, both Dewey and Jack seem to be the most expressive member of their groups, group A and B, respectively. As can be observed, both of them have the highest lines of their respective group in all the four emotions considered.

In addition, it is interesting to observe that the levels of "joy" and "anger" of both groups, A and B, is higher than their levels of "fear" and "sadness". From that, we could infer that the participants of both groups felt fine, they did not feel under pressure and, somehow, enjoyed themselves.

VI. CONCLUSION AND FUTURE WORK

In this work, we aimed at presenting an experiment on semantic emotional evaluation of chats. There are already some previous works in semantic emotional evaluation, as the ones mentioned in Section II, but they differ from Chat-SEE goals in several senses.

On one hand, Chat-SEE makes use of a different emotion classification, which, though taken from the psychological research area [12], has been re-interpreted in order to be used in our chat environment.

On the other hand, we were mainly interested in the emotion evolution from a relative point of view; that is: the emotion evolution among members of a group which were faced to work out a task collaboratively. So, we put more emphasis on the conclusions that could be derived within each group, rather than on the individual scores.

In that sense, Chat-SEE has obtained interesting results, because we have been able to measure how emotions evolve in an electronic conversation, being able to somehow "quantify" how they evolve. Moreover, Chat-SEE seems to be able to identify some kind of leadership role within conversations, as could be the case with Dewey and Jack. Exploring that possibility also is part of our future work.

There are some other challenges we face after this experiment.

Firstly, it is clear that the emotional dictionary used becomes a key module in the process, given that a bad emotional dictionary would clearly bias the final results. In

 ISBN: 978-1-61208-175-5

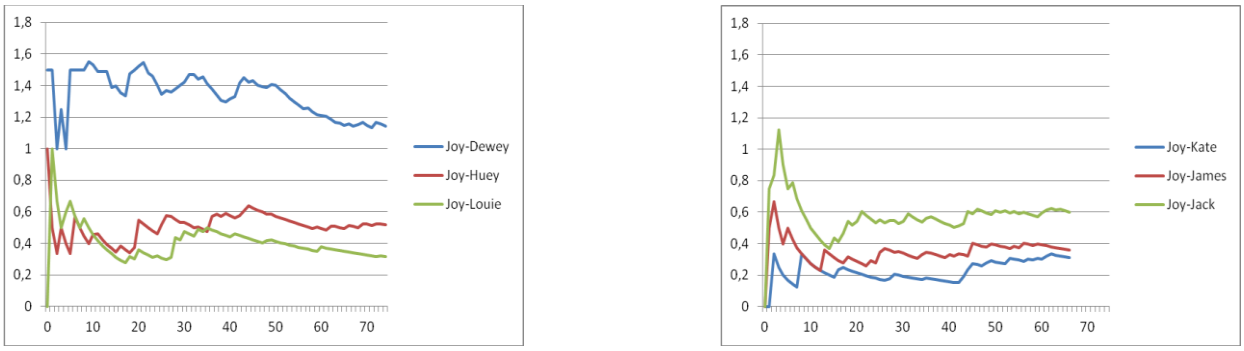


Figure 3. "Joy" representation for groups A and B.

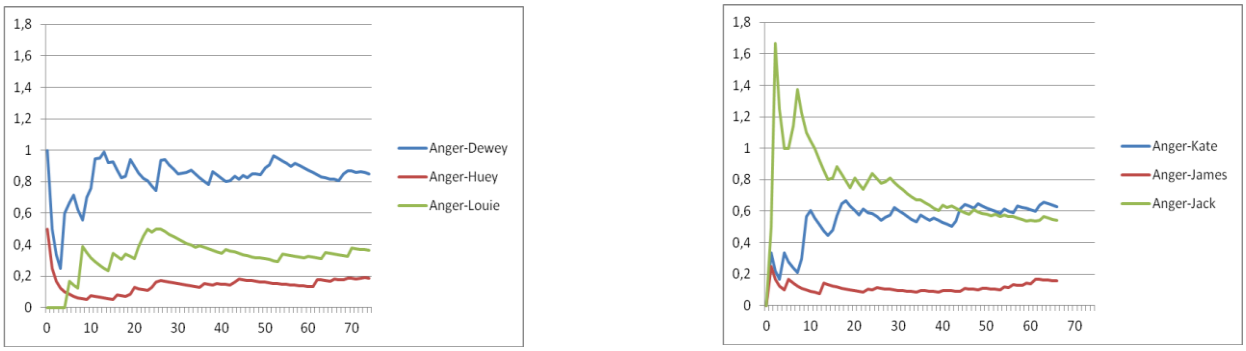


Figure 4. "Anger" representation for groups A and B.

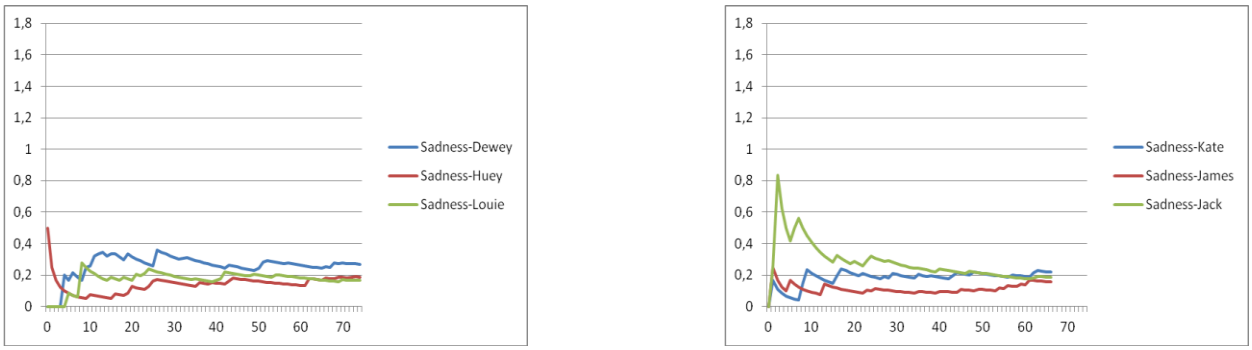


Figure 5. "Sadnes" representation for groups A and B.

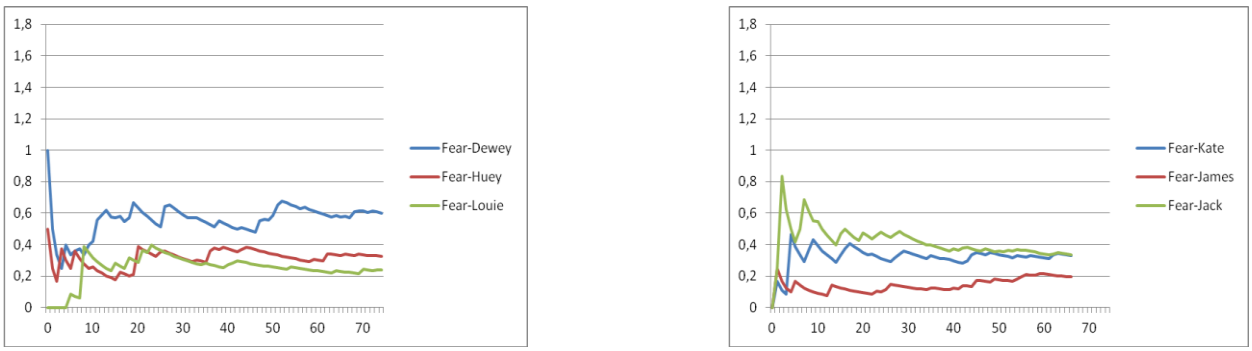


Figure 6. "Fear" representation for groups A and B.

 ISBN: 978-1-61208-175-5

that sense, we are aimed at improving the dictionary in two ways:

a) by including some kind of natural language preprocessing before the semantic emotion annotation,

b) by stablishing a judge protocol that would validate the semantic emotion asignment.

Moreover, the accumulation algorithm used has also become as a key module. We could modify our algorithm in several ways: media per paragraph, etc. Also, we could modify different parameters, as well as the weight given to them, by assigning different weight to the emotional dimensions depending on the chat subject. A comparative human analysis of the emotions of the chat is foreseen, in order to evaluate the correctness of the evaluation.

Finally, we plan to develop a graph zoom to be used for zooming instant peaks, and implement an online evaluator integrated in a chat tool. That online evaluator would let supervisors to react if some situations are identified.

ACKNOWLEDGMENT

This work has been partially supported by the Spanish Ministry of Science and Education (TIN2010-17344) and Comunidad Autonoma de Madrid (S2009/TIC-1650).

REFERENCES

[1] Ex-Sense, <http://www.ex-sense.com/proversion.html> 09.26.2011

[2] Á. Rodríguez Bravo et al., "Modelización acústica de la expresión emocional en el español." Procesamiento del lenguaje natural, nº. 25, Sept. 1999, pp. 159-166.

[3] Glad or sad, <http://www.gladorsad.com/en/>09.26.2011

[4] MultimediaN, Emotional Analyzer, <http://www.multimedian.nl/en/demo_emotional_analyzer.php>09.26.2011

[5] A. Friberg, "A fuzzy analyzer of emotional expression in music performance and body motion", Proc. Music and Music Science, 2004, pp. 1-13.

[6] C. Strappavana and R. Mihalcea, "Learning to identify emotions in text," Proc. 23th ACM Symposium on Applied Computing (SAC'08), ACM Press, 2008, pp. 1556-1560, , doi:10.1145/1363686.1364052.

[7] Headline Analyzer, <http://www.aminstitute.com/headline/> 09.26.2011

[8] SAS ® Sentiment analysis, <http://www.sas.com/text-analytics/sentiment-analysis/> 09.26.2011

[9] S. Asur and B. A. Huberman, "Predicting the Future with Social Media", Proc. WI-IAT'10, 2010 IEEE/WIC/ACM I nternational Conference on Web Intelligence and Intelligent Agent Technology, IEEE Computer Society, 2010, doi: 10.1109/WI-IAT.2010.63

[10] M. Taboada, J. Brooke, M. Tofiloski, K. Voll and M. Stede, "Lexicon-Based Methods for Sentiment Analysis", Computational Linguistics, vol. 37:2, Jun. 2011, pp. 267-307, doi:10.1162/COLI_a_00049

[11] A. Kalra and K. Karahalios, "TextTone: expressing emotion through text." Interact 2005, LNCS 3585, Springer, 2005, pp. 966-969, doi: 10.1007/11555261_81

[12] A. Zinck and A. Newen, "Classifying emotion: a developmental account." Synthese, vol 161:1, Jan. 2008, pp. 1-25, doi: 10.1007/s11229-006-9149-2

[13] D. Mertz, "Charming Python: Get Started with the Natural Language Toolkit.", 2004, <http://www.ibm.com/developerworks/linux/library/l-cpnltk.html > 09.20.2011

[14] Natural language toolkit, <http://www.nltk.org/> 09.20.2011

 ISBN: 978-1-61208-175-5