IARIA

SEMAPRO 2010

The Fourth International Conference on Advances in Semantic Processing

October 25-30, 2010 - Florence, Italy

Editors

Manuela Popescu

Darin L. Stewart

SEMAPRO 2010

Foreword

The Fourth International Conference on Advances in Semantic Processing (SEMAPRO 2010), held from October 25 to October 30, 2010 in Florence, Italy, considered the complexity of understanding and processing information. 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.

With the progress on ontology, web services, semantic social media, semantic web, deep web search /deep semantic web/, semantic deep web, semantic networking and semantic reasoning, SEMAPRO 2010 constituted the stage for the state-of-the-art on the most recent advances.

We take here the opportunity to warmly thank all the members of the SEMAPRO 2010 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 that dedicated much of their time and efforts to contribute to the SEMAPRO 2010. We truly believe that thanks to all these efforts, the final conference program consists of top quality contributions.

This event could also not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the SEMAPRO 2010 organizing committee for their help in handling the logistics and for their work that is making this professional meeting a success. We gratefully appreciate to the technical program committee co-chairs that contributed to identify the appropriate groups to submit contributions.

We hope Florence provided a pleasant environment during the conference and everyone saved some time for exploring this historic city.

SEMAPRO 2010 Chairs

Harith Alani, Knowledge Media Institute/The Open University, UK
Petre Dini, IARIA / Concordia University, Canada
Laurianne Sitbon, National ICT Australia/Queensland Research Laboratory - Brisbane, Australia
René Witte, Concordia University - Montréal, Canada
Stefania Galizia, INNOVA S.p.A., Italy
Peter Haase, Fluid Operations, Germany
Thorsten Liebig, derivo GmbH - Ulm, Germany
Nima Dokoohaki, Royal Institute of Technology (KTH)-Kista, Sweden
Ralf Krestel, L3S Research Center - Hannover, Germany

Massimo Paolucci, DOCOMO Communications Laboratories Europe GmbH - Munich, Germany
Darin L. Stewart, Oregon Health & Science University, USA

SEMAPRO 2010

Committee

SEMAPRO Advisory Chairs
Harith Alani, Knowledge Media Institute/The Open University, UK
Petre Dini, IARIA / Concordia University, Canada
Laurianne Sitbon, National ICT Australia/Queensland Research Laboratory - Brisbane, Australia
René Witte, Concordia University - Montréal, Canada

SEMAPRO 2010 Industry Liaison Chairs
Stefania Galizia, INNOVA S.p.A., Italy
Peter Haase, Fluid Operations, Germany
Thorsten Liebig, derivo GmbH - Ulm, Germany

SEMAPRO 2010 Research/Industry Chairs
Nima Dokoohaki, Royal Institute of Technology (KTH)-Kista, Sweden
Ralf Krestel, L3S Research Center - Hannover, Germany
Massimo Paolucci, DOCOMO Communications Laboratories Europe GmbH - Munich, Germany
Darin L. Stewart, Oregon Health & Science University, USA

SEMAPRO 2010 Technical Program Committee
Nasser Alalwan, King Saud University - Riyadh, Saudi Arabia
Harith Alani, Knowledge Media Institute/The Open University, UK
José F. Aldana Montes, University of Málaga, Spain
Sören Auer, AKSW-Universität Leipzig, Germany
Sofia J. Athenikos, Drexel University, USA
Ebrahim Bagheri, National Research Council, Canada
Sean Bechhofer, University of Manchester, UK
Eva Blomqvist, STLab/ISTC-CNR, Italy
Janez Brank, Jozef Stefan Institute - Ljubljana, Slovenia
Ozgu Can, Ege University, Turkey
Stefania Costache, University of Hannover, Germany
Anne Cregan, Intersect Australia, Australia
Mathieu d'Aquin, KMi/The Open University, UK
Violeta Damjanovic, Salzburg Research - Salzburg, Austria
Hendrik Decker, Valencia Polytechnic University, Spain
Jan Dedek, Charles University in Prague, Czech Republic
Alexiei Dingli, The University of Malta, Malta
Nima Dokoohaki, Royal Institute of Technology (KTH)-Kista, Sweden
Jiri Dokulil, Charles University in Prague, Czech Republic
Raimund Ege, Northern Illinois University, USA
Anna Fensel, FTW Forschungszentrum Telekommunikation Wien GmbH, Austria
Stefania Galizia, INNOVA S.p.A., Italy
Raúl García Castro, Universidad Politécnica de Madrid, Spain
Fabio Grandi, University of Bologna, Italy
Carole Goble, Manchester University, UK

Sotirios K. Goudos, Aristotle University of Thessaloniki, Greece
Daniela Grigori, University of Versailles, France
Stephan Grimm, FZI - Research Center for Information Technologies Karlsruhe, Germany
Alessio Gugliotta, Innova SpA, Italy
Peter Haase, Fluid Operations, Germany
Ralf Heese, Freie Universität Berlin, Germany
Christian F. Hempelmann, RiverGlass, Inc., USA / Purdue University, USA
Wladyslaw Homenda, Warsaw University of Technology, Poland
Vasant Honavar, Iowa State University -Ames, USA
Carolina Howard Felicissimo, Schlumberger SITS Norge AS - Oslo, Norway
Zhisheng Huang, Vrije Unversity Amsterdam, The Netherlands
Prasad Jayaweera, University of Namur, Belgium
Najla Sassi Jaziri, ISSAT Mahdia, Tunisia
Wassim Jaziri, ISIM Sfax, Tunisia
Ivan Jelinek, Czech Technical University - Prague, Czech Republic
Jana Katreniakova, Comenius University Bratislava, Slovakia
Ralf Krestel, L3S Research Center - Hannover, Germany
Petr Kroha, TU Chemnitz, Germany
Kyu-Chul Lee, Chungnam National University, South Korea
Sang-goo Lee, Seoul National University, Korea
Thorsten Liebig, derivo GmbH - Ulm, Germany
Christian Meilicke, University Mannheim, Germany
Paola Mello, DEIS - University of Bologna, Italy
Elisabeth Métais, CNAM/CEDRIC, France
Vasileios Mezaris, Informatics and Telematics Institute - Centre for Research and Technology Hellas (ITI-CERTH), Greece
Ekawit Nantajeewarawat, SIIT-Thammasat University, Thailand
Lyndon J. B. Nixon, STI International, Austria
Massimo Paolucci, DOCOMO Communications Laboratories Europe GmbH - Munich, Germany
Carlos Pedrinaci, Knowledge Media Institute / The Open University, UK
Andrea Perego, Università degli Studi dell'Insubria - Varese, Italy
Jaime Ramírez, 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
Marco Ronchetti, Università di Trento, Italy
Michele Ruta, Politecnico di Bari, Italy
Melike Sah, University of Dublin, Ireland
Satya Sahoo, Wright State University, USA
Munehiko Sasajima, ISIR / Osaka University, Japan
Minoru Sasaki, Ibaraki University, Japan
Christoph Schmitz, 1&1 Internet AG - Karlsruhe, Germany
Laurianne Sitbon, National ICT Australia/Queensland Research Laboratory - Brisbane, Australia
William Song, Durham University, UK
Darin L. Stewart, Oregon Health & Science University, USA
Umberto Straccia, ISTI - CNR, Italy

Cui Tao, Mayo Clinic, USA
Merwyn G. Taylor, MITRE Corporation, USA
Saïd Tazi, LAAS-CNRS, Université de Toulouse / Université Toulouse1, France
Christoph Tempich, Detecon International GmbH - Bonn   Germany
Leonardo Vito, University of Camerino, Italy
Tomas Vitvar, University of Innsbruck, Austria
Johanna Voelker, University of Mannheim, Germany
Roland Wagner, University of Linz - Austria
Haofen Wang, Shanghai Jiao Tong University, China
Shenghui Wang, Vrije Universiteit Amsterdam, The Netherlands
René Witte, Concordia University - Montréal, Canada
Wai Lok Woo, Newcastle University - Newcastle upon Tyne, UK
Xian Wu, IBM China Research Lab - Beijing, P.R.C.
Filip Zavoral, Charles University in Prague, Czech Republic
Nadia Zerida, Paris 8 University, France
Yuting Zhao, The University of Aberdeen, UK
Hai-Tao Zheng, Tsinghua University, China

Copyright Information

Table of Contents

Similarity Features, and their Role in Concept Alignment Learning

Shenghui Wang*, Gwenn Englebienne†, Christophe Guéret*, Stefan Schlobach*, Antoine Isaac*, Martijn Schut*
*Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands
†Informatics Institute, Universiteit van Amsterdam, The Netherlands
Email:{swang,cgueret,schlobac,aisaac}@few.vu.nl, schut@cs.vu.nl, G.Englebienne@uva.nl

***Abstract*—Finding mappings between compatible ontologies is an important and difficult open problem. Instance-based methods for solving this problem have the advantage of focussing on the most active parts of the ontologies and reflect the semantics of the ontologies as they are used in the real world. We evaluate how the feature representation of the instances is representative of the corresponding concepts, investigate how this corresponds with the domain characteristics of the data and which role it plays in the task of instance-based ontology mapping. We use two different competitive classifiers and a standard feature selection to identify important features, and study the effect of those different classifiers in the concept alignment context.**

***Keywords*-Instance-based Ontology Matching, Semantic Interoperability, Machine Learning**

I. INTRODUCTION

Motivation: The problem of semantic heterogeneity and the resulting problems of interoperability and information integration have been studied for well over 40 years now. It is at present an important hurdle to the realisation of the Semantic Web. Solving matching problems is one step to the solution of the interoperability problem. Semantic Web community has invested significant efforts over the past few years [1].

Solving the matching problem requires to assess the conceptual similarity between elements of two separate ontologies in order to determine relationships (mappings) such as equivalence or subsumption between them. One way of judging whether two concepts from different ontologies are semantically equivalent is to observe their *extensional* information, that is, the instance data they classify [2], [3], [4]. However, it is not always easy to identify identical instances in many applications. Therefore, a robust instance-based mapping technique should cope with the case when there are no explicitly common instances.

Problem Description: This paper focus on instance-based mapping technique only. In [5], we formulated the matching problem as a classification problem, where a mapping can be predicted from the similarity between the extensional information of two concepts.

As in many other application contexts, the instances are described and can be compared according to many dimensions (*features*). Knowing which of these features play the most important role during the classification is important as to optimise the quality of meta-data. Important features would be taken more care of. It is thus interesting to look for a way of assessing the relative importance of the features. In this paper, we use two different automated methods, namely Markov Random Field (MRF) and Evolution Strategy (ES) to investigate this importance. Concept mapping can be seen as a side effect of these methods, and the quality of the method can be assessed by the quality of the concept mapping it produces. We therefore also compare the concept-mapping performance of our methods to a state-of-the-art, off-the-shelf classifier: the Support Vector Machine (SVM).

Research Questions: Our aim is to answer the following research questions:

- What are the benefits of using a machine learning algorithm to determine the importance of features?
- Are there regularities *wrt.* the relative importance given to specific features for similarity computation? Are these weights related to application data characteristics?
- How do different classifiers perform on this instance-based mapping task?

Findings: The two classifiers provide largely consistent, sensible and valuable insight in the importance of the instance features. As evaluated against a human golden standard, they also outperform the SVM on the concept mapping task, thereby indicating that the highlighted features are indeed important.

II. PROBLEM STATEMENT

Our task is to match two thesauri, GTT and Brinkman, which are used to annotate different book collections at the National Library of the Netherlands (Koninklijke Bibliotheek or KB). In order to improve the interoperability between these collections, for example, using GTT concepts to search books annotated only with Brinkman concepts, we need to find mappings between these two thesauri.

As investigated in [3], books annotated by a concept can be treated as instances of this concept. Using shared book instances has already provided interesting mappings. However, many books are not used, because they are not dually annotated. In this paper, we further our investigation in [5], focus on finding mappings directly using book meta-data, no matter the books are dually annotated or not.

Books are described by their title, author, abstract, *etc.* These features together represent an individual book instance. For each concept, all its instances are grouped into an integrated representation of this concept, feature by feature. For example, all titles of these books are put together as a "bag of words." Term frequencies are measured within bags, so that a concept is represented by a high dimensional vector where each element represents the frequency of a term. The similarity between two concepts is calculated with respect to each feature, using the cosine similarity between the term frequencies in these bags.

The similarity between the two elements of a pair of concepts i is therefore measured and represented by a high dimensional vector F_i. The similarity between feature j of the concepts is indicated by F_{ij}. These similarity vectors can be treated as points in a space. In this "similarity space," each dimension corresponds to the similarity in terms of one feature. As we know, some points (*i.e.*, some pairs of concepts) are real mappings but some are not. Our hypothesis is that the *label* of a point — whether it represents a mapping or not — is correlated with the position of this point in this space.

Given some existing mappings, *e.g.*, from a manual effort, our goal is to *learn* this correlation. Therefore the mapping problem is transformed into a classification problem. With already labelled points and the actual similarity values of concepts involved, it is possible to classify a point — *i.e.*, to decide whether the pair represents a mapping — based on its location in the similarity space. One baseline method is to apply a standard support vector machine (SVM) to find a hyperplane which separates classes with different labels. Another option is to look for a direct correlation between labels and similarities. Here we adopt two classifiers: one based on a graphical Markov Random Field [6] and the other using multi-objective evolution strategy [7].

III. METHODS

All three methods assume that mappings are independent. This is a simplifying assumption (since if a term A maps to B, the probability that A maps to any $C \neq B$ clearly decreases), but it is necessary because explicitly modelling the dependencies between all possible mappings is intractable.

A. Markov Random Field (MRF)

Let $T = \{\, (F_i, L_i) \,\}_{i=1}^{N}$ be the training set of mappings, with, for each given pair of concepts i, a feature vector $F_i \in \mathbb{R}^K$, where K is the number of features, and an associated label L_i.

We consider a simple graphical model, consisting of an observed multivariate input F and a latent variable L which represents the label. We assume that the mappings are identically distributed conditionally on the observations, and model the conditional probability of a mapping given the input, $p(L_i|F_i)$, using a probability distribution from the exponential family. That is:

$$p(L_i|F_i) = \frac{1}{Z(F_i)} \exp\big(\sum_{j=1}^{K} \lambda_j f_j(L_i, F_i)\big), \tag{1}$$

where $\Lambda = \{\, \lambda_j \,\}_{j=1}^{K}$ are the weights associated to the potential function and $Z(F_i)$, called the partition function, is a normalisation constant ensuring that the probabilities sum to 1. It is given by

$$Z(F_i) = \sum_{L \in \{0,1\}} \exp\big(\sum_{j=1}^{K} \lambda_j f_j(L, F_i)\big). \tag{2}$$

Because of our assumption that mappings are independent, the likelihood of the data set for given model parameters $p(T|\Lambda)$ is given by:

$$p(T|\Lambda) = \prod_{i=1}^{N} p(L_i|F_i) \tag{3}$$

During learning, our objective is to find the most likely values for Λ. We assume a *prior probability* distribution on Λ which favours small values, assigning a normal distribution with zero mean and covariance σ^2 for each λ_i. The *posterior* probability of Λ is then given by

$$p(\Lambda|T) = p(T|\Lambda)p(\Lambda)/p(T), \tag{4}$$

where $p(T)$ is a normalisation term which does not depend on Λ and can therefore be ignored during optimisation. Moreover, since the logarithm is a monotonically increasing function, we can optimise $\log p(\Lambda|T)$ rather than $p(\Lambda|T)$; this turns out to be easier. Ignoring constants, the function we optimise is thus:

$$\ell(\Lambda) = \sum_{i=1}^{N} \left[\sum_{j=1}^{K} \lambda_j f_j(L_i, F_i) - \log Z(F_i) \right] - \sum_{j=1}^{K} \frac{\lambda_j^2}{2\sigma^2}. \tag{5}$$

This is equivalent with logistic regression, where we assume a linear function for the discriminant and introduce regularisation on the model parameters. The result is a convex function which can easily be optimised using any variation of gradient ascent. We used the L-BFGS [8] for the results presented here. The first derivative of $\ell(\Lambda)$ is given by

$$\sum_{i=1}^{N} \left[f_j(L_i, F_i) - \sum_{L \in \{0,1\}} f_j(L, F_i) p(L|F_i, \Lambda) \right] - \frac{\lambda_j}{\sigma^2} \tag{6}$$

The variance of the prior, σ, is a parameter that has to be set by hand and can be seen as a regularisation parameter which prevents overfitting of the training data. The decision criterion for assigning a label to a new pair of concepts is then given by:

$$L_i^P = \begin{cases} 1 & \text{if } p(L_i = 1|F_i) > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

B. *Multi-Objective Evolution Strategy*

The evolutionary computing paradigm consists of a number of algorithms (genetic algorithms, evolutionary programming, and others) that are based on, among others, natural selection and genetic inheritance; these algorithms are used for optimisation, modelling and simulation. For the purpose of this paper, we decided to use evolutionary strategies (ES). Evolutionary strategies have two characteristic properties: firstly, they are used for *continuous value* optimisation, and, secondly, they are *self-adaptive*. The first property is desirable for our problem at hand, because we are dealing with real-valued representations. The second property makes the search strategy adaptive, *i.e.*, it dynamically changes search parameters if necessary. Such self-adaptation is shown to be highly effective in complex search processes where it is difficult to tune the parameters manually.

As compared with the genotype/phenotype solution encoding used in Genetic Algorithm, an ES individual is a direct model of the searched solution. That is, an individual is defined by Λ and some evolution strategy parameters:

$$\langle \Lambda, \Sigma \rangle \leftrightarrow \langle \lambda_1, \dots, \lambda_K, \sigma_1, \dots, \sigma_K \rangle \tag{8}$$

Then, a metric for the quality of individuals — a fitness function — is established. The fitness function is related to the decision criterion for the ES, which is sign-based:

$$L_i^{ES} = \begin{cases} 1 & \text{if } \sum_{j=1}^{K} \lambda_i F_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

From 9, we can see that maximising the number of positive results and negative results are two opposite goals. Those goals can be expressed as a multi-objective fitness function using a first component f_1 for the number of true positives matches and the other one f_2 for the number of true negatives.

$$\begin{aligned} f_1(\Lambda \mid F, L) &= \#\{F_i \mid \sum_{j=1}^{K} \lambda_i F_{ij} > 0 \wedge L_i = 1\} \quad (10) \\ f_2(\Lambda \mid F, L) &= \#\{F_i \mid \sum_{j=1}^{K} \lambda_i F_{ij} \leq 0 \wedge L_i = 0\} \quad (11) \end{aligned}$$

Instead of searching for one global optimum, this definition allows the finding of best compromises between errors made on positive and negatives matches.

The evolution process itself essentially consists of three operators: the recombination, mutation and survivor selection operators.

- Recombination is applied on two parent individuals $\langle \lambda_1^1, \dots, \lambda_K^1, \sigma_1^1, \dots, \sigma_K^1 \rangle$ and $\langle \lambda_1^2, \dots, \lambda_K^2, \sigma_1^2, \dots, \sigma_K^2 \rangle$. From an arithmetic recombination weighted by a coefficient γ, a first new individual $\langle \lambda'_1, \dots, \lambda'_K, \sigma'_1, \dots, \sigma'_K \rangle$ is created:

$$\begin{aligned} \lambda'_j &= (1-\gamma_j)\lambda_j^1 + \gamma_j \lambda_j^2, \quad j = 1, \dots, K \quad (12) \\ \sigma'_j &= (1-\gamma_j)\sigma_j^1 + \gamma_j \sigma_j^2, \quad j = 1, \dots, K \quad (13) \end{aligned}$$

similarly, an second child $\langle \lambda''_1, \dots, \lambda''_K, \sigma''_1, \dots, \sigma''_K \rangle$ is created with $\sigma''_j = \gamma_j \sigma_j^1 + (1-\gamma_j)\sigma_j^2$ and $\lambda''_j = \gamma_j \lambda_j^1 + (1-\gamma_j)\lambda_j^2$. The value of γ is drawn from a uniform distribution on $[0, 1]$.

- Mutation is applied on one parent individual $\langle \lambda_1, \dots, \lambda_K, \sigma_1, \dots, \sigma_K \rangle$. It results in the creation of one child $\langle \lambda'_1, \dots, \lambda'_K, \sigma'_1, \dots, \sigma'_K \rangle$.

$$\begin{aligned} \lambda'_j &= \lambda_j + \sigma'_j \mathcal{N}_j(0,1), \quad j = 1, \dots, K \quad (14) \\ \sigma'_j &= \sigma_j \exp^{\tau' \mathcal{N}(0,1) + \tau \mathcal{N}_j(0,1)}, \quad j = 1, \dots, K \quad (15) \end{aligned}$$

with $\mathcal{N}(0,1)$ being a random number drawn from a "standard" normal distribution (i.e. with mean equal to 0 and standard deviation of 1). The notation $\mathcal{N}_j(0,1)$ denotes the use of a different value for every j^{th} strategy parameter. The two τ parameters are used to define a learning rate. Following conventions, we set them to be inversely proportional to the square root of problem size $\tau = 1/\sqrt{2\sqrt{K}}$ and $\tau' = 1/\sqrt{2K}$.

- Survivor selection is performed using the NSGA2 [9] strategy. The parent population and the offspring solution are joined into one unique, temporary, population. Those individuals are sorted into different fronts according to Pareto optimality. Starting form the best non dominated front of solution, each successive front is made of next non dominated solution that are not yet in a front. Those fronts are used to generate the new parent population. When not all the elements in a front can be picked up, the selection between the individuals in such a way it preserves diversity.

During one loop of the algorithm, new candidate solutions are created using recombination and/or mutation until an oversize criterion is reached. Then, survivor operator is applied to lower the number of individuals to the original population size. The final result of the learning process is the set of best solutions found, according to Pareto optimality. An expert can use the system, stop it at any time and pick up a solution among the best ones found so far. In the absence of an expert, a simple heuristic is used: The winner is the individual whose positive score is the closest to the average of positives scores for all the population. We implemented the ES classifier using OpenBeagle [10], keeping a population of 30 individuals at each iteration.

C. *Support Vector Machine*

Support vector machines (SVMs) are a set of machine learning algorithms classically used for classification and regression problems [11]. Our work concerns the assessment of a mapping for a given similarity vector. That is, binary classification. In this context, SVM can be used as a maximum margin classifier whose task consists in finding an hyperplane h, with parameters $\omega \in \mathbb{R}^K$ and $b \in \mathbb{R}$, separating the two classes. A sign-based criterion allows the attribution of a class $c_i \in \{-1, +1\}$ to a data vector i.

$$c_i = \begin{cases} +1 & \text{if } \langle \omega \cdot F_i \rangle + b > 0 \\ -1 & \text{if } \langle \omega \cdot F_i \rangle + b \leq 0 \end{cases} \tag{16}$$

The objective is to maximise the margin separating the two classes whilst minimizing classification error risk. Classification is expressed as a constraint. The decision rule from the equation 16 can be changed into the constraint in equation 17 (where N is the number of elements in the training dataset).

$$c_i(\langle \omega \cdot F_i \rangle + b) \geq 1, \quad i = 1, \ldots, N \tag{17}$$

The margin to maximize separates each class set of points closest to the hyperplane. Those support vectors satisfy the condition $|| \langle \omega \cdot F_i \rangle + b ||_2 = 1$. It can be shown that maximizing this margin is equivalent to minimizing the quantity $\frac{1}{2}\langle \omega \cdot \omega \rangle$.

We now have an objective to minimize and some constraints. Next step of SVM formulation is to take the Lagrangian $\mathcal{L}(\omega, \alpha, b)$ of this optimisation problem. This notation introduces a set of Lagrange coefficients $\alpha_i \in \mathbb{R}^+$.

$$\mathcal{L}(\omega, \alpha, b) = \frac{1}{2}\langle \omega \cdot \omega \rangle - \sum_{i=1}^{N} \alpha_i [c_i(\langle \omega \cdot F_i \rangle + b) - 1] \tag{18}$$

This formulation is only able to deal with data that is strictly linearly separable. In order to deal with non linearly separable datasets, the scalar product $\langle F_i \cdot F_j \rangle$ is replaced by a kernel function $K(F_i, F_j)$. The expected outcome of this so called "kernel trick" is to map the data from $\mathbb{R}^K$ to a higher dimension space were they will be linearly separable. Moreover, a tolerance for error is added by setting a maximum boundary C for the α_i. The final optimization problem is:

$$\begin{array}{ll} Max. & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j c_i c_j K(F_i, F_j) \\ with & \sum_{i=1}^{N} \alpha_i c_i = 0 \\ and & 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, N \end{array} \tag{19}$$

And the final decision criterion for the SVM is:

$$L_i^{SVM} = \begin{cases} 1 & \text{if } \sum_{l=1}^{N} \alpha_l c_l K(F_l, F_i) + b \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

The choice of the kernel function has a sensitive impact on the performance of the classifier. Practically, it dictates the shape of the surface that will surround the two classes. We decided to use the commonly used Radial Basis Function (RBF) to get "potato-shaped" classes. This kernel is expressed as

$$K(F_i, F_j) = \exp\left(-\gamma || F_i - F_j ||_2\right). \tag{21}$$

We used the implementation of libSVM for the results reported here, with $\gamma = 0.5$ and $C = 8$.

λ_j	Feature	λ_j	Feature	λ_j	Feature
1	Lexical	11	author	21	issued
2	Jaccard	12	contributor	22	language
3	Date	13	creator	23	mods:edition
4	ISBN	14	dateCopyrighted	24	publisher
5	NBN	15	description	25	refNBN
6	PPN	16	extent	26	relation
7	SelSleutel	17	hasFormat	27	spatial
8	abstract	18	hasPart	28	subject
9	alternative	19	identifier	29	temporal
10	annotation	20	isVersionOf	30	title

Table I
LIST OF THE FEATURES

IV. EXPERIMENTS

We match the GTT and Brinkman thesauri, which contain 35K and 5K concepts respectively. They are used to annotate two book collections of the KB, containing 2M books of which nearly 1M books were annotated, including 307K books with GTT concepts only; 490K with Brinkman concepts only; 222K with both.

A. Feature selection for similarity calculation

On top of the similarity calculated using book metadata, as introduced in Section II, we also measured the relative edit distance as the lexical distance between two concepts. The Jaccard similarity measure used in [3] is also included. Note that the Jaccard measure is calculated from dually annotated books only. If two concepts are never used to annotate dually indexed books, we set the Jaccard measure to be the average of all calculated Jaccard measures. The features used are listed in Table I and all similarity values are normalised to have zero mean and unit variance in order to make comparison of λ_i meaningful.

The lexical and Jaccard similarity are of course strong indicators of concept mappings, and may seem to give artificially high results for our instance-based method. However, it is a great advantage that we can include any information in the features, and let the machine decide on their relative importance. For reference, Figure 1 includes how the MRF performs when these two features are removed ("MRF 3-30"). It shows that we still obtain quite good results from the instances only, although the best results are obtained with the combination ("MRF 1-30").

B. Control-Experiment: Quality of Learning

First, we used human labelled pairs to carry on 10-fold cross validation in order to check validity of our learned mappings. These pairs of concepts were judged by a human evaluator who assigned a "mapping" or "non-mapping" label to each pair of concepts. The similarity between these pairs of concepts were calculated as introduced above. The whole data set was divided into 10 folds, each time using 9 folds to train the probabilistic model and the remaining fold to test the model.

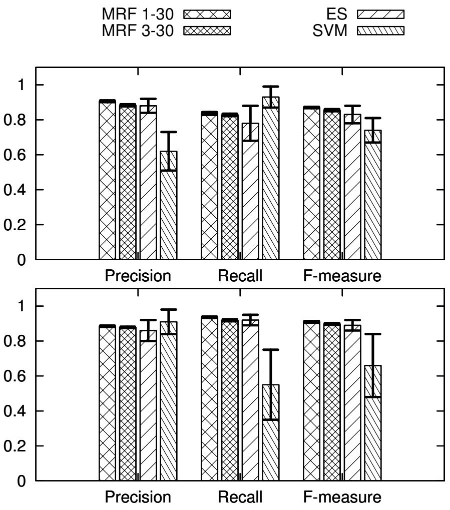


Figure 1. Precision, recall and F-Measure for mappings with a positive label (top) and a negative label (bottom). Error bars indicate one standard deviation over the 10 folds of cross-validation.

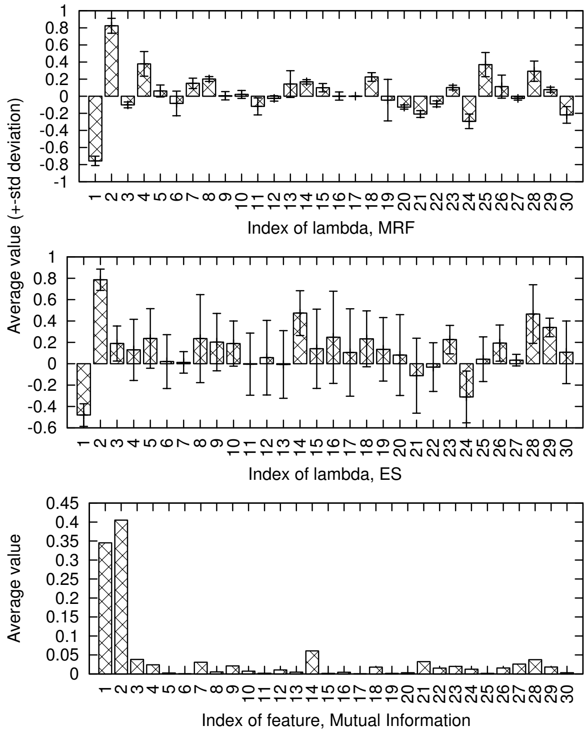


Figure 2. Values of Λ and mutual information between features and labels

In the testing step, the predicted mappings were compared with the real mappings. The positive precision is the proportion of real mappings among all predicted positive mappings, and the positive recall is the proportion of true predicted mappings among all real mappings. The negative precision is the proportion of the non-mappings among all predicted negative mappings and the recall is the proportion of the predicted non-mappings among all non-mappings. Figure 1 shows the performance of the three classifiers. These show to be generally quite good for the MRF and ES methods, with performances comparable to the results of state-of-the-art mappers [12]. Our deployment of SVM generally performs worse than MRF and ES. One possible reason for this may be the tuning of the parameters γ and C. Another reason may be our choice of the RBF kernel which is perhaps not optimal for this problem. However, those results clearly show that our chosen classifier are highly competitive and perform favorably wrt. state of the art matching tools.

C. Relative importance of features

An important benefit of our first two methods is that the solutions are interpretable by humans. In an attempt to work out which features of our instances are important for mapping, we explored whether the value of λ_i reflects the intuitive importance of feature i. Figure 2 depicts how the weights (the values of λ) varied over the 10 folds of cross-validation for the MRF and ES classifiers, as well as the mutual information between the mapping label and each similarity feature.

A first observation is that ES lambdas are not really conclusive: the 10 solutions are much less consistent than MRF ones. Reassuringly, however, ES lambdas that are most inconclusive correspond to the least informative features (as shown by the mutual information). Focusing on the MRF, then, we can observe that apart from a few exceptions, important features in terms of mutual information are associated to large weights, while unimportant features are normally associated to small weights. Notice in that respect that feature 1 is a distance measure, while all other features are similarity measures. Some less informative features still have large weights (*e.g.*, feature 25), however. This may be explained by the fact that the mutual information was computed independently for all features. A feature may be completely random overall, yet be informative conditionally on some other feature. The combination of such features will still be informative and result in larger weights. Similarly, a feature may be very informative by itself, yet not provide any supplementary value (and may even be detrimental) if another feature already provides the same information, thus explaining some features with high mutual information have low weights.

A more detailed examination of the weights allows us to compare the learnt importance of features with the intuitions provided by the application context. A first set of features has large weights as expected, such as the similarity between the concept labels (feature 1), their co-occurrence in the set of dually-annotated books (feature 2) and the subject (feature 28). A few features are expected not to play a significant role and have indeed low importance: size of the book (16), (rare) format description (17) and language (22), for instance.

Some features, more surprisingly, were given an importance level that conflicts with what one could have anticipated: description (15) and abstract (8), which give readable descriptions of book content, happen to be only marginally important, less than for example the date of copyright (14). The latter, for instance, may mirror phenomena like the publication of a number of books on the same subject in short periods of time or, perhaps, that some concepts are used a lot for a short period, and much less before and after that period.

This last category illustrates how learning can help making decisions in dubious cases. For instance, it is well known that book titles (30) do not always cover their subject entirely. Our experiments demonstrate that similarity between these rather hints at conceptual dissimilarity — even though this is less clear for the alternative titles (9). Similarly, two books may refer to different subjects while being written by the same author(s). This is especially true when homonymy is not dealt with. — creator, author, contributors, respectively 11, 12, 13 — or published by the same publisher (24).

This observation tends to show that when many different description features interact, there is no systematic correlation between what a learning method could find and what an application expert may anticipate. And in such cases it is highly valuable, for tuning mappers exploiting instance similarities, to apply learning techniques instead of relying solely on human judgement.

V. Conclusion

In this paper, we take the instance-based mapping technique one step further and investigate what instance features are important in this context. Our analysis has shown that the overall similarity of instances is too coarse a measure: the similarity of some features is very indicative of a valid mapping while some are not and, even worse, the similarity of some instance features actually indicates concept dissimilarity.

Two different machine learning techniques are used to automatically identify meaningful features. Both methods assign mostly consistent importance to the features, which agrees with the domain characteristics of the data.

The two classifiers we propose, the MRF and the ES, result in a performance in the neighbourhood of 90%, showing the validity of the approach. Their performance is not significantly different, but both significantly outperform the SVM, an off-the-shelf classifier.

In the future, we would like to investigate how instance similarity can be used to infer multi-concept mappings (n to m mappings). We would also like to learn the type of mapping (for example “broader than,” “narrower than,” as defined in the SKOS standard [13]), using multiple labels in the classification process.

References

[1] J. Euzenat and P. Shvaiko, *Ontology Matching*. Springer Verlag, 2007.

[2] R. Ichise, H. Takeda, and S. Honiden, “Integrating multiple internet directories by instance-based learning,” *Proceedings of the eighteenth International Joint Conference on Artificial Intelligence*, 2003.

[3] A. Isaac, L. van der Meij, S. Schlobach, and S. Wang, “An empirical study of instance-based ontology matching,” in *Proceedings of the 6th International Semantic Web Conference*, Busan, Korea, 2007.

[4] C. Wartena and R. Brussee, “Instanced-based mapping between thesauri and folksonomies,” in *Proceedings of the 7th International Semantic Web Conference*, Karlsruhe, Germany, 2008.

[5] S. Wang, G. Englebienne, and S. Schlobach, “Learning concept mappings from instance similarity,” in *Proceedings of the 7th International Semantic Web Conference*, Karlsruhe, Germany, 2008.

[6] R. Kindermann and J. L. Snell, *Markov Random Fields and Their Applications*. AMS, 1980.

[7] H.-G. Beyer and H.-P. Schwefel, “Evolution strategies: A comprehensive introduction,” *Journal Natural Computing*, vol. 1, no. 1, p. 352, 2002.

[8] D. C. Liu and J. Nocedal, “On the limited memory method for large scale optimization,” *Mathematical Programming*, vol. 45, pp. 503–528, 1989.

[9] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II,” in *Proceedings of the Parallel Problem Solving from Nature VI Conference*. Paris, France: Springer. Lecture Notes in Computer Science No. 1917, 2000, pp. 849–858.

[10] C. Gagn and M. Parizeau, “Genericity in evolutionary computation software tools: Principles and case-study,” *International Journal on Artificial Intelligence Tools*, vol. 15, no. 2, pp. 173–194, April 2006.

[11] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, March 2000.

[12] C. Caracciolo, J. Euzenat, L. Hollink, R. Ichise, A. Isaac, V. Malaisé, C. Meilicke, J. Pane, P. Shvaiko, H. Stuckenschmidt, O. Sváb-Zamaza, and V. Svátek, “Results of the ontology alignment evaluation initiative,” Tech. Rep., 2008.

[13] A. Isaac and E. Summers, “Skos primer,” Working Draft, W3C, Tech. Rep., March 17 2009. [Online]. Available: http://www.w3.org/TR/skos-primer/

SOIT: Spatial Ontologies Integration Tool and its Application to the Road Domain

Wassim Jaziri, Sana Châabane, Faïez Gargouri

Miracl Laboratory, ISIM Sfax,
Pôle Technologique, Route de Tunis Km 10,
B.P. 242, SFAX 3021, Tunisia
Emails: (jaziri.wassim, sana.chaabane, faiez.gargouri)@gmail.com

***Abstract*—Designed as semantic structures to support the sharing and reuse of geographic data, spatial ontologies have recently gained attention within the geo-information community. Geographic ontologies are designed to provide a common understanding of the structure of geographic models, and to support the development of geographic information systems that are conceptually complex. This paper proposes an approach for merging spatial ontologies based on three complementary modules: matching, mapping and merging. A Spatial Ontologies Integration Tool (SOIT) is also developed and applied to the road domain.**

Keywords-Spatial ontologies; SOIT; Integration tool; Geographic Information systems; Road domain.

I. INTRODUCTION

In the applications with spatial vocation (e.g., geographic domain), ontologies are an effective solution in particular to ensure interoperability and semantic cooperation between Geographic Information systems (GIS). Spatial ontologies offer a relevant solution for the sharing and the integration of geographic data.

The problem of heterogeneity of geographic ontologies is more complex than that of other domain ontologies [5]; because it is necessary to take into account the spatial and temporal aspects as well as rules governing the data evolution.

Spatial data interoperability allows simplifying and enhancing the sharing, reuse and integration of geographic data. However, semantic heterogeneity [8] is a major obstacle to the interoperability of geographic data [9]. Indeed, the implementation of a geographic ontology can manage and structure multiple data sets that can be grouped according to geographical criteria. Its objective is not to only describe the list of existing geographic objects (territory, boundary, road network, etc.) but to identify classes; to define the relationships may exist between them, and to describe the attributes in order to obtain the knowledge base.

In this paper, we aim to resolve the problem of heterogeneity of geographic ontologies by proposing a merging approach.

This paper is organized as follows: Section 2 presents an overview about tools and techniques for merging ontologies. The Third Section details the proposed approach for merging geographical ontologies. A spatial ontologies integration tool is described in Section 4. Section 5 presents the application of our approach and tool on the road domain. The conclusion and outlook of our work are listed in Section 6.

II. ONTOLOGY MERGING TOOLS AND TECHNIQUES

Several ontologies merging tools were developed in literature, such as [6]: Chimaera, FCA-Merge, PROMPT, OntoMorph and ONIONS.

- Chimaera: It is an interactive ontologies merging tool that allows the diagnosis, the test and the edition of the merging result [4]. It helps user to find the best term by proposing a list of the used terms while helping to resolve the terminological difficulties. This tool, based on the Ontolingua ontology editor, offers a support for the merging process to enable the collection of ontologies expressed in different formalisms. It makes the translation at the language level and uses heuristics to find the parts of the ontology to be reorganized.

- PROMPT: Based on a semi-automatic merging approach, it allows making certain tasks automatically and helps the user along the merging process [5]. PROMPT determines possible filminesses in the state of the ontology resulting from user's actions and suggest solutions for them.

- OntoMorph is based on a merging approach which is similar to the two previous tools [2]. An expert uses an initial list of correspondences between concepts of the source ontologies: the user defines a set of operators that are applied to ontologies for resolving inconsistencies.

- FCA-Merge: It uses a formal, bottom-up method of ontology merging based on the extraction of concepts from textual documents [7]. It applies natural language processing and generates a "concept trellis" from “FCACore” algorithm. This trellis is transformed subsequently into domain ontology by an expert of the domain.

- ONIONS (ONtological Integration Of Naive Sources) is a method designed for the conceptual analysis and ontological integration of terminologies [3]. This method consists of two steps: (1) A reengineering step which consists in the extraction, formatting, analysis and formalization of data; (2) A merging step which allows the merging of ontologies using an algorithm based on algebra.

The developed tools do not consider the spatial aspect of objects describing the geographical domain. The spatial dimension as well at the intrinsic level of the concepts at the level of the spatial and semantic relationships were lacking to these tools. This limits their applicability in geographical ontologies.

III. SPATIAL ONTOLOGIES MERGING APPROACH

We developed an approach for geographical ontologies merging, based on two criteria:

- *The identity search*: the search for relationships of spatial identity and total identity between concepts of the initial ontologies.

Definition 1. Two objects are spatially identical if they are located in the same place but having a different characteristic such as the instance name or the acquisition date.

Definition 2. Two objects are totally identical if they are spatially and semantically identical. By semantic identity, we mean that both objects have the same name and the same properties.

This criterion allows obtaining the skeleton of the ontology result of merging process. We thus join the not identical individuals of the candidate ontologies to serve as entries to the second step of the merging process.

- *The search for enrichment relationships*. Enrichment relationships have two types: the semantic relationships such as *equivalence* and *part-of* and the spatial relationships such as *adjacency, intersection, joint, junction* etc.

The proposed approach is based on three main modules: (1) matching module, (2) mapping module and (3) merging module (Figure 1). The first module consists in determining the matching process between candidate ontologies. The output of this phase is a list of matching functions. The second phase allows finding correspondences between concepts of candidate ontologies. The result of this phase is two lists: a list of matches between candidate concepts and a second list of concepts without correspondences. The third phase is merging which is based on merging rules. It produces as result a comprehensive ontology spatially and semantically richer than the candidate ontologies.

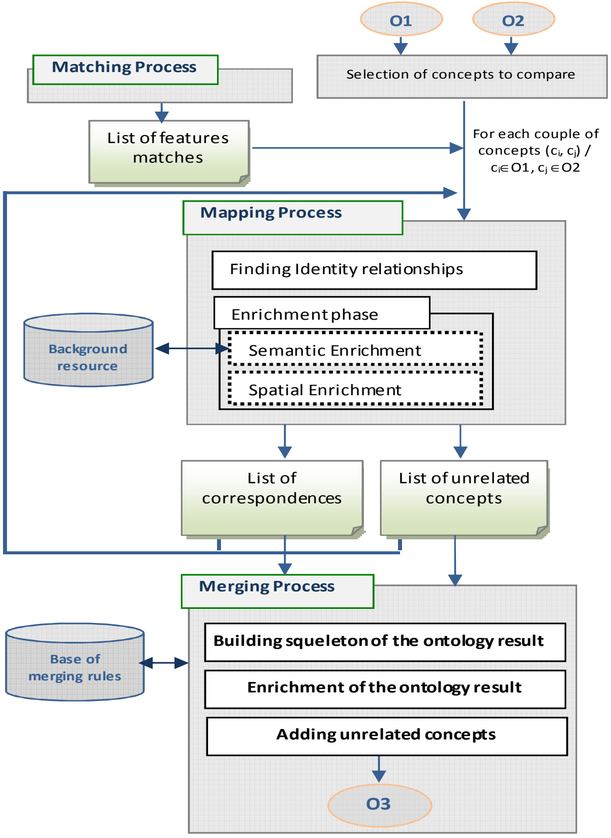

O1
O2
Matching Process
Selection of concepts to compare
List of features matches
For each couple of concepts (cᵢ, cⱼ) / cᵢ∈O1, cⱼ∈O2
Mapping Process
Finding Identity relationships
Enrichment phase
Background resource
Semantic Enrichment
Spatial Enrichment
List of correspondences
List of unrelated concepts
Merging Process
Base of merging rules
Building squeleton of the ontology result
Enrichment of the ontology result
Adding unrelated concepts
O3

Figure 1. The proposed approach for merging spatial ontologies.

3.1 The matching phase

A matching process defines a set of functions which specifies correspondences between terms of ontologies. This phase gives as a result a list of features matches. There are two types of relationships considered in our approach:

- Connecting relationships: they are the relating points between two ontologies. We distinguish spatial identity relationships and semantic identity relationships.

- Enriching relationships: we distinguish semantic relationships and topological relationships (intersection, union, etc.).

We use two types of matching: spatial and semantic matching. In the semantic matching, we define two functions: the first one defines semantic identity relationship (Idsem) and the second function defines semantic enrichment relationships between candidate concepts. In the spatial matching, we define two functions: the spatial identity relationship (Idspa) and the spatial enrichment relationships.

The semantic Identity: Idsem means that two concepts have the same name and same properties. We use the syntactic technique to derive such relationship. We use the edit distance of Levenshtein to calculate the similarity between concepts names and their properties. This measure of edit distance ed represents the minimum number of insertions, deletions or substitutions necessary to transform one string x into another y. Similarity s(x, y) normalized to [0,1] is defined as follows:

s=1-ed/max(| x |,| y |).

We consider that two concepts C1 and C2 admit a semantic identity relationship Idsem(C1, C2) if and only if:

s(C1.name,C2.name) = 1

and for every attribute atti of C1, there exists an attribute attj of C2 where s(C1.atti, C2.attj)=1, and vice versa, for every attribute of C2, there is an attribute of C1 where s(C2.attj,C1.atti) = 0.

The spatial identity relationship Idspa relates only to geographical concepts. A spatial object is described according to its graphical form: GF (point, line or polygon), semantics data (eg name, nature, appearance, various characteristics) and localization data (position on the surface). The search for Idspa relationship between concepts of candidate ontologies consists in comparing localization characteristics of concepts.

We formally define the relationships considered by our approach. We present the following formal definitions defined.

Definition 1: Two objects are spatially identical if they are located in the same place but having a different characteristic such as the instance name or the acquisition date.

Definition 2: Two objects are totally identical if they are spatially and semantically identical.

Let us consider the concepts C1(X1, Y1) and C2(X2, Y2) with X1, Y1 and X2, Y2 are coordinates of C1 and C2 respectively and C1.GF=point and C2.GF=point. We consider Idspa(C1, C2) if and only if Euclidean distance dE(C1,C2) =0. The function of spatial identity relationship is defined as follows:

$$\mathbf{IDspa{:}=}\left\{\begin{matrix}\mathbf{C1\in O1, C2\in O2\wedge C1.FG=point\ \wedge}\\ \mathbf{C2.FG=point\wedge dE=0}\end{matrix}\right\}. \qquad (1)$$

$$\mathbf{dE=\sqrt{(x1-x2)^2+(y1-y2)^2}.} \qquad (2)$$

3.2 The mapping phase

The input of this phase is two concepts from both candidate ontologies. The mapping process is iterative and consists of two steps. The first step is to investigate Identity relationships and the second step is to investigate Enrichment relationships between candidate concepts. The search for enrichment relationships is performed on non-identical concepts selected at the previous phase. The compared concepts and their correspondences are stored in a base of matches. Concepts which the mapping algorithm found no connections between them, i.e. the concepts that do not verify any type of relationship between them (called unrelated concepts) are stored in a base of unrelated concepts.

We have defined rules to optimize the number of comparisons of concepts in order to avoid a randomly process. For example, to research identity relationship, we rely on the type of the graphic form of the concept to make comparisons.

3.3 The merging phase

The merging phase consists in building the ontology result. The input of this phase is composed of bases of correspondences and unrelated concepts. The aim of this module is to apply the correspondence links stored at the correspondences bases (semantic and geographical) in accordance with merging techniques. The merging process creates a new geographical ontology from two candidate ontologies connected by identity concepts which are used as connected points between the two ontologies and enriched by the semantic and geographical relationships.

Rules for merging candidate concepts are defined. These rules are of two types: rules for semantic relationships and rules for spatial relationships. The merging rules are applied to concepts accepting connections between them. Unrelated concepts are transmitted in the ontology result without any treatment.

IV. SOIT: Spatial Ontologies Integration Tool

We have developed the SOIT tool (Spatial Ontologies Integration Tool) (Figure 2) based on Java language and the integrated development environment (IDE) NetBeans. The tool is designed for automatically merge two spatial ontologies. SOIT takes as input two spatial ontologies written in OWL and produces as result an ontology spatially and semantically richer. In addition, SOIT allows other functions: it can perform two types of matching candidate ontologies and see (2 the result of the matching process. It can also generate)the graph of an ontology written in OWL and view or print one ontology in the form of a text or a graph. We model a use case diagram of UML language representing the various functionalities of the SOIT tool (Figure 2).

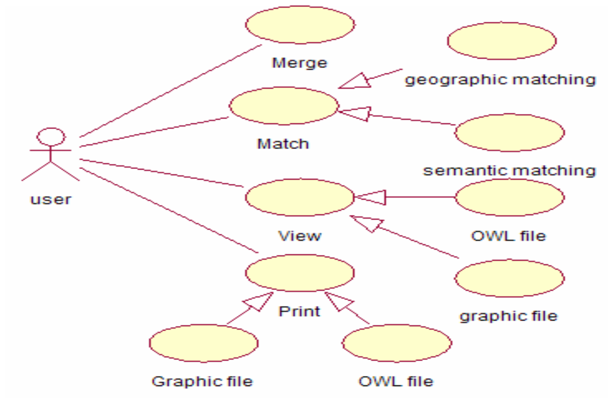
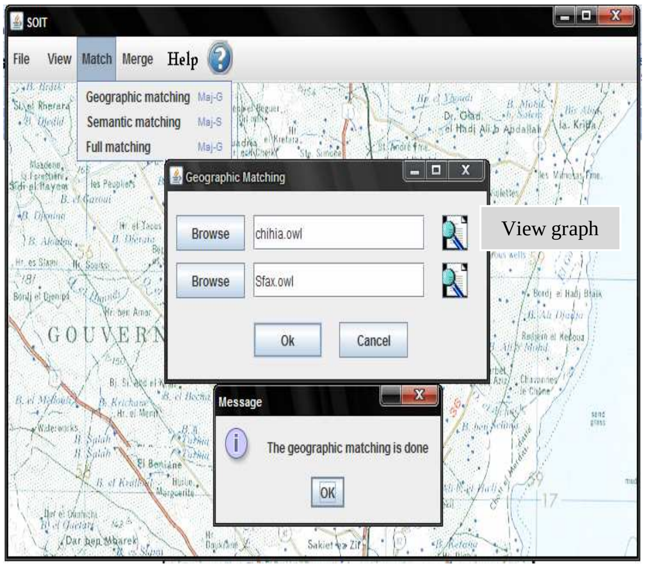


Figure 2. The Use Case diagram of SOIT.

The host interface of SOIT includes a menu bar contains five menus: “File”, “View”, “Match”, “Merge” and “Help”.

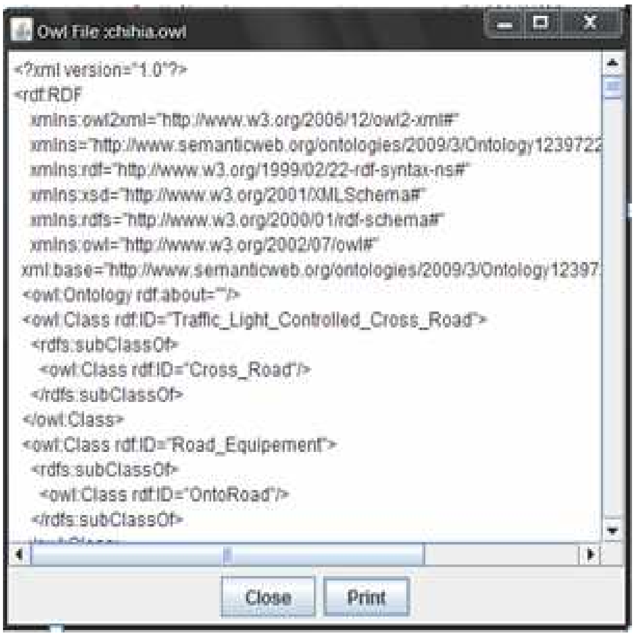
The matching process starts with the introduction of two candidate ontologies (figure 3).



The geographic matching is done

Figure 3. The functionalities of the "Match" menu.

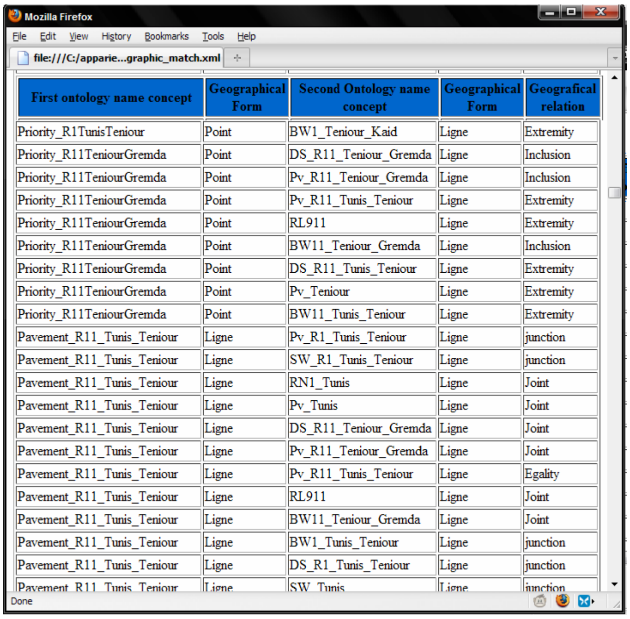
The graph and the OWL file of a candidate ontology can be viewed through the button "view graph" (figures 4).



```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns="http://www.semanticweb.org/ontologies/2009/3/Ontology1239722
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
 xml:base="http://www.semanticweb.org/ontologies/2009/3/Ontology12397
 <owl:Ontology rdf:about=""/>
 <owl:Class rdf:ID="Traffic_Light_Controlled_Cross_Road">
  <rdfs:subClassOf>
   <owl:Class rdf:ID="Cross_Road"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Road_Equipement">
  <rdfs:subClassOf>
   <owl:Class rdf:ID="OntoRoad"/>
  </rdfs:subClassOf>
```

Figure 4. The OWL file of the candidate ontology.

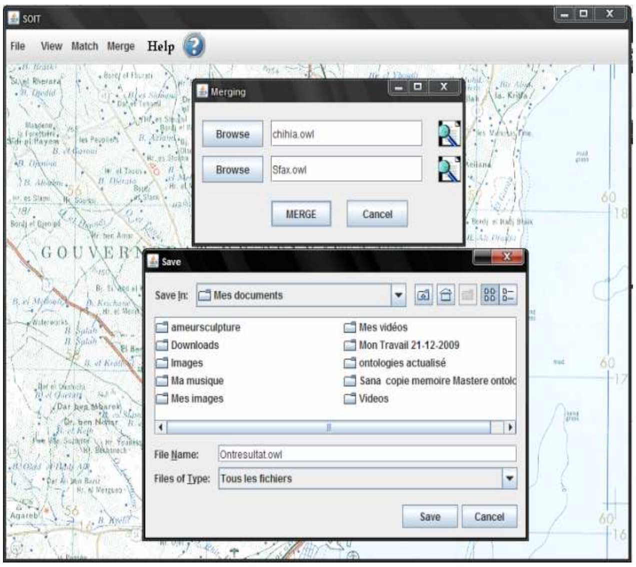
After running the matching process, the system displays the list of matches found. This functionality is performed using XSLT style sheets (figure 5). For example, we have identified a relationship of type *Extremity* between individuals: *Priority_R1_Tunis_Teniour* and *BW1_Teniour_Kaied*.



First ontology name concept	Geographical Form	Second Ontology name concept	Geographical Form	Geografical relation
Priority_R1TunisTeniour	Point	BW1_Teniour_Kaid	Ligne	Extremity
Priority_R11TeniourGremda	Point	DS_R11_Teniour_Gremda	Ligne	Inclusion
Priority_R11TeniourGremda	Point	Pv_R11_Teniour_Gremda	Ligne	Inclusion
Priority_R11TeniourGremda	Point	Pv_R11_Tunis_Teniour	Ligne	Extremity
Priority_R11TeniourGremda	Point	RL911	Ligne	Extremity
Priority_R11TeniourGremda	Point	BW11_Teniour_Gremda	Ligne	Inclusion
Priority_R11TeniourGremda	Point	DS_R11_Tunis_Teniour	Ligne	Extremity
Priority_R11TeniourGremda	Point	Pv_Teniour	Ligne	Extremity
Priority_R11TeniourGremda	Point	BW11_Tunis_Teniour	Ligne	Extremity
Pavement_R11_Tunis_Teniour	Ligne	Pv_R1_Tunis_Teniour	Ligne	junction
Pavement_R11_Tunis_Teniour	Ligne	SW_R1_Tunis_Teniour	Ligne	junction
Pavement_R11_Tunis_Teniour	Ligne	RN1_Tunis	Ligne	Joint
Pavement_R11_Tunis_Teniour	Ligne	Pv_Tunis	Ligne	Joint
Pavement_R11_Tunis_Teniour	Ligne	DS_R11_Teniour_Gremda	Ligne	Joint
Pavement_R11_Tunis_Teniour	Ligne	Pv_R11_Teniour_Gremda	Ligne	Joint
Pavement_R11_Tunis_Teniour	Ligne	Pv_R11_Tunis_Teniour	Ligne	Egality
Pavement_R11_Tunis_Teniour	Ligne	RL911	Ligne	Joint
Pavement_R11_Tunis_Teniour	Ligne	BW11_Teniour_Gremda	Ligne	Joint
Pavement_R11_Tunis_Teniour	Ligne	BW1_Tunis_Teniour	Ligne	junction
Pavement_R11_Tunis_Teniour	Ligne	DS_R1_Tunis_Teniour	Ligne	junction
Pavement_R11_Tunis_Teniour	Ligne	SW_Tunis	Ligne	junction

Figure 5. Geographic matching result.

For merging two ontologies, the user has to introduce two geographic ontologies instances of the same model, by clicking on the button “Browse”. The following window displays (Figure 6).



File Name: Ontresultat.owl

Files of Type: Tous les fichiers

Figure 6. Selection of a candidate ontology.

Finally, by clicking on the button "Merge", the user can visualize the concepts and the individuals of the ontology result (figure 7).

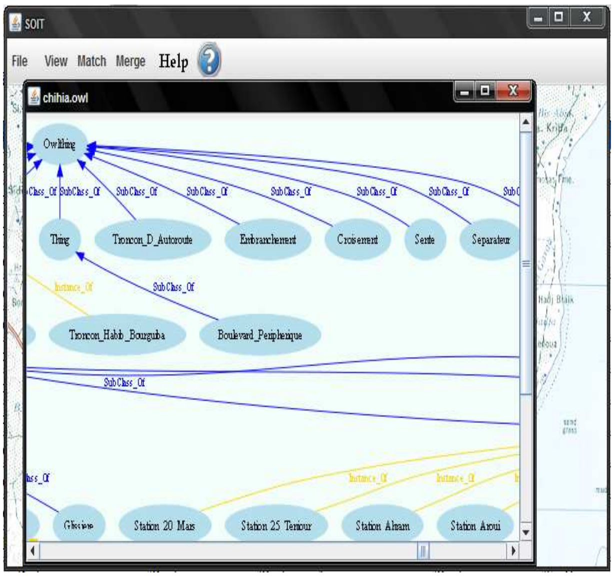


Figure 7. Graph of the ontology result.

V. APPLICATION TO THE ROAD DOMAIN

The application domain of the developed spatial ontologies integration tool is the road domain. We developed two spatial ontologies related to the city of Sfax (Tunisia), called respectively *ontoRoadChihia.owl* and *ontoRoadSfax.owl* instances of the OntoRoad ontology [1] which is developed to model the road domain concepts .

The studied corpus is composed of topographic maps. The instantiation of the *OntoRoad* ontology is made by geographical zone. Both candidate ontologies subject of experiment cover different geographical zones from the city of *Sfax* (Tunisia). We extract all the objects of the considered zone and we attribute them to their corresponding classes. For example, the object *Hedi_Chaker* is a Street; the street *Ibn_kholdoun* is one-way.

The following extract presents the modelling of the individual "*RL911*" of the concept "*Local_Road*" in the ontology *ontoRoadChihia.owl* (Table I).

TABLE I. EXTRACT OF THE ONTOLOGY ONTOROADCHIHIA.OWL.

```
<Local_Road rdf:about="#RL911">
         <Position_Route rdf:datatype="&xsd;string">0.0</Position_Route>
         <Debut_De_Section_Voie rdf:datatype="&xsd;string"
               >Carrefour_G_3Chemins</Debut_De_Section_Voie>
         <Fin_De_Section_Voie
rdf:datatype="&xsd;string">Km12</Fin_De_Section_Voie>
         <Forme_geometrique
rdf:datatype="&xsd;string">Ligne</Forme_geometrique>
         <Nom_Route
rdf:datatype="&xsd;string">Route_Teniour</Nom_Route>
                <Rencontre_Voie_Voie rdf:resource="#Av_7Novembre"/>
         <A_Droite_De rdf:resource="#Av_7Novembre"/>
         <Rencontre_Voie_Voie rdf:resource="#Av_Afrique"/>
         <A_Droite_De rdf:resource="#Av_Afrique"/>
         <A_Droite_De rdf:resource="#Av_Teboulbi"/>
               <A_Droite_De rdf:resource="#BW11_Teniour_Gremda"/>
         <Rencontre_Voie_Voie
rdf:resource="#BW11_Teniour_Gremda"/>
                <A_Gauche_De rdf:resource="#BW11_Tunis_Teniour"/>
         <Rencontre_Voie_Voie rdf:resource="#BW11_Tunis_Teniour"/>
            <Rencontre_Voie_Voie rdf:resource="#BW1_Teniour_Kaid"/>
         <A_Droite_De rdf:resource="#BW1_Teniour_Kaid"/>
         <Rencontre_Voie_Voie rdf:resource="#BW1_Tunis_Teniour"/>
         <A_Gauche_De rdf:resource="#BW1_Tunis_Teniour"/>
         <A_Gauche_De rdf:resource="#RL921"/>
         <Rencontre_Voie_Voie rdf:resource="#RL921"/>
               <Adjacence_Route_Trottoir rdf:resource="#SW_Teniour"/>
                <Rencontre_Voie_Voie rdf:resource="#S_Khaledwalid"/>
         <A_Droite_De rdf:resource="#S_Khaledwalid"/>
         <Rencontre_Voie_Voie rdf:resource="#S_Tina"/>
         <A_Droite_De rdf:resource="#S_Tina"/>
</Local_Road>
```

The following extract presents the modelling of the same individual in the *ontoRoadSfax.owl* (Table II).

TABLE II. EXTRACT OF THE ONTOLOGY ONTOROADSFAX.OWL.

```
<Local_Road rdf:about="#RL911">
         <Position_Route rdf:datatype="&xsd;string">0.0</Position_Route>
         <Debut_De_Section_Voie rdf:datatype="&xsd;string"
              >Carrefour_G_3Chemins</Debut_De_Section_Voie>
         <Fin_De_Section_Voie
rdf:datatype="&xsd;string">Km12</Fin_De_Section_Voie>
         <Forme_geometrique
rdf:datatype="&xsd;string">Ligne</Forme_geometrique>
         <Nom_Route
rdf:datatype="&xsd;string">Route_Teniour</Nom_Route>
         <Rencontre_Voie_Voie rdf:resource="#Av_5Aout"/>
             <Rencontre_Voie_Voie rdf:resource="#Av_Majida_Boulila"/>
               <A_Droite_De rdf:resource="#BW11_Teniour_Gremda"/>
         <Rencontre_Voie_Voie
rdf:resource="#BW11_Teniour_Gremda"/>
                <A_Gauche_De rdf:resource="#BW11_Tunis_Teniour"/>
         <Rencontre_Voie_Voie rdf:resource="#BW11_Tunis_Teniour"/>
            <Rencontre_Voie_Voie rdf:resource="#BW1_Teniour_Kaid"/>
                  <A_Droite_De rdf:resource="#BW1_Teniour_Kaid"/>
         <Rencontre_Voie_Voie rdf:resource="#BW1_Tunis_Teniour"/>
                 <A_Gauche_De rdf:resource="#BW1_Tunis_Teniour"/>
         <Connexion_Extremite-Noeud rdf:resource="#GCR_3Chemins"/>
         <A_Droite_De rdf:resource="#RL_Kaid"/>
         <A_Gauche_De rdf:resource="#RN1_Tunis"/>
               <Adjacence_Route_Trottoir rdf:resource="#SW_Teniour"/>
</Local_Road>
```

The following extract presents the result of merging of these two ontologies (Table III).

TABLE III. EXTRACT OF THE ONTOLOGY RESULT.

```
<Local_Road rdf:about="#RL911">
        <Position_Route rdf:datatype="&xsd;string">0.0</Position_Route>
                <Debut_De_Section_Voie rdf:datatype="&xsd;string"
            >Carrefour_G_3Chemins</Debut_De_Section_Voie>
        <Fin_De_Section_Voie
         rdf:datatype="&xsd;string">Km12</Fin_De_Section_Voie>
        <Forme_geometrique
rdf:datatype="&xsd;string">Ligne</Forme_geometrique>
        <Nom_Route
rdf:datatype="&xsd;string">Route_Teniour</Nom_Route>
        <Rencontre_Voie_Voie rdf:resource="#Av_5Aout"/>
             <Rencontre_Voie_Voie rdf:resource="#Av_7Novembre"/>
        <A_Droite_De rdf:resource="#Av_7Novembre"/>
        <Rencontre_Voie_Voie rdf:resource="#Av_Afrique"/>
        <A_Droite_De rdf:resource="#Av_Afrique"/>
           <Rencontre_Voie_Voie rdf:resource="#Av_Majida_Boulila"/>
        <A_Droite_De rdf:resource="#Av_Teboulbi"/>
             <A_Droite_De rdf:resource="#BW11_Teniour_Gremda"/>
        <Rencontre_Voie_Voie
rdf:resource="#BW11_Teniour_Gremda"/>
              <A_Gauche_De rdf:resource="#BW11_Tunis_Teniour"/>
        <Rencontre_Voie_Voie rdf:resource="#BW11_Tunis_Teniour"/>
           <Rencontre_Voie_Voie rdf:resource="#BW1_Teniour_Kaid"/>
                <A_Droite_De rdf:resource="#BW1_Teniour_Kaid"/>
        <Rencontre_Voie_Voie rdf:resource="#BW1_Tunis_Teniour"/>
              <A_Gauche_De rdf:resource="#BW1_Tunis_Teniour"/>
        <Connexion_Extremite-Noeud rdf:resource="#GCR_3Chemins"/>
        <A_Gauche_De rdf:resource="#RL921"/>
        <Rencontre_Voie_Voie rdf:resource="#RL921"/>
        <A_Droite_De rdf:resource="#RL_Kaid"/>
        <A_Gauche_De rdf:resource="#RN1_Tunis"/>
             <Adjacence_Route_Trottoir rdf:resource="#SW_Teniour"/>
              <Rencontre_Voie_Voie rdf:resource="#S_Khaledwalid"/>
        <A_Droite_De rdf:resource="#S_Khaledwalid"/>
        <Rencontre_Voie_Voie rdf:resource="#S_Tina"/>
        <A_Droite_De rdf:resource="#S_Tina"/>
</Local_Road>
```

VI. CONCLUSION AND FUTURE WORK

The need to combine ontologies developed in an independent way and containing heterogeneity, raised problems from the point of view of the ontological language, the conceptualization and the specification. The heterogeneity between the knowledge expressed within each of the ontologies treating the same domain must be resolved. Several solutions to produce much more successful ontologies were proposed and varied techniques were developed for the adaptation, the merging and the integration. The integration is the construction of a new ontology reusing the other available ontologies which will be a part of the new ontology. The logical integration of two ontologies supplies to the user a vision unified by various sources.

In this paper, we have presented an approach for merging geographic ontologies. This approach consists of three processes: (1) the matching process, (2) the mapping process and (3) the merging process. We also developed SOIT: a tool for spatial ontologies integration. The application of this tool has been made on the road domain. Our ongoing work are to evaluate "SOIT" by comparing the result produced by this tool with the one developed by an expert in the field. In future work, we aim at extending this tool with functionalities for query ontological data bases.

REFERENCES

[1] Chaabane, S. and Jaziri, W., OntoRoad : Spatial ontology for the road domain, Journées Francophones sur les Ontologies (JFO'2008), pages 135-144, December1-2, 2008, Lyon, France.

[2] Chalupsky, H., OntoMorph: A translation System For Symbolic Kcowledge, In proceedings of the 7th international conference on principles of knowledge representation and reasoning (KR'2000), pages 471-482, Colorado, USA, 2000.

[3] Gangemi, A., Pisanelli, D.M. and Steve, G., Ontology Integration: Experience with medical terminologies, In N. Guarino (ed.), Formal Ontology in Information Systems, pages 163-178. IOS Press, 1998.

[4] McGuinness, D.L., Fikes, R., Rice, J. and Wilder, S., The Chimaera ontology Environment. In proceeding of the seventeenth national conference on artificial intelligence (AAAI 2000), 2000.

[5] Noy, N.F. and Musen, M.A., PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In proceeding of the 17th national conference on artificial intelligence (AAAI 2000), 2000.

[6] Poulain, T. and Cullot, N., Ontology Mappings for Information Systems Cooperation. Deuxièmes Journées Francophone sur les Ontologies (JFO2008), ACM Edition, pages 47-52, 2008.

[7] Stumme, G. and Maedche, A., FCA-merge: Bottom-Up Merging of Ontologies, Proceedings of the International Conference on Artificial Intelligence (IJCAI'01), pages 225-230, USA, 2001.

[8] Villie M. and Félix S., Ontologies: solving semantic heterogeneity in a federated spatial database system. Proceeding of 5th International Conference on Enterprise Information System, pages 347-352, Angers, France, April 2003.

[9] Sheth A.P. and Larson J.A.., Federated database systems for managing distributed heterogeneous and autonomous databases. ACM Computing Surveys, 1990.

Ontology Alignment Technique for Improving Semantic Integration

Mohammad Mustafa Taye
Faculty of Information Technology
Philadelphia University, Jordan
mtaye@philadelphia.edu.jo

Nasser Alalwan
Software Technology Research Laboratory (STRL),
De Montfort University
Leicester, UK line
nasser@dmu.ac.uk

Abstract— **A new technique for ontology alignment has been built by integrating important features of matching to achieve high quality results when searching and exchanging information between ontologies. The system is semi-automatic and enables syntactical and semantic interoperability among ontologies. Moreover, it is a multi-strategy algorithm which can deal with and solve more than one critical problem. Therefore, it is likely to be more conveniently applicable in different domains. Also, we improve a semantic matcher based on combining lexical matcher with several rules and facts. Moreover, our technique illustrates the solving of the key issues related to heterogeneous ontologies, which uses combination-matching strategies to execute the ontology-matching task. Therefore, it can be used to discover the matching between ontologies. The main aim of the work is to introduce a method for finding semantic correspondences among heterogeneous ontologies, with the intention of supporting interoperability over given domains. Our goal is to achieve the highest number of accurate matches.**

Keywords-Ontology; Semantic Interoperability; Heterogeneous; Ontology Alignment.

I. INTRODUCTION

Ontology [1] has been developed to offer a commonly agreed understanding of a domain that is required for knowledge representation, knowledge exchange and reuse across domains. Therefore, ontology organizes information into taxonomies of terms (i.e., concepts, attributes) and shows the relationships between them. In fact, it is considered to be helpful in reducing conceptual confusion for users who need to share applications of different kinds, so it is widely used to capture and organize knowledge in a given domain.

Although ontologies are considered to provide a solution to data heterogeneity, from another point of view, the available ontologies could themselves introduce heterogeneity problems.

In order to deal with these problems, ontologies must be available for sharing or reusing; therefore, semantic heterogeneity and structural differences need to be resolved among ontologies. This can be done, in some cases, by aligning or matching heterogeneous ontologies. Thus, establishing the relationships between terms in the different ontologies is needed throughout ontology alignment [4, 5, 7, 14].

Semantic interoperability can be established in ontology reconciliation. The original problem is called the “ontology alignment”. The alignment of ontologies is concerned with the identification of the semantic relationships (subsumption, equivalence, etc.) that hold between the constituent entities (which can be classes, properties, etc.) of two ontologies.

In this paper, an ontology alignment technique has been developed in order to facilitate communication and build a bridge between ontologies. An efficient mechanism has been developed in order to align entities from ontologies in different description languages (e.g., OWL, RDF) or in the same language. This approach tries to use all the features of ontologies (concept, attributes, relations, structure, etc.) in order to obtain efficiency and high quality results. For this purpose, several matching techniques have been used such as string, structure, heuristic and linguistic matching techniques with thesaurus support, as well as human intervention in certain cases, to obtain high quality results. This paper is organized as follows: section II over view about our system; Section III describes our system in details. Section IV and Section V shows the related work and the evaluation process. Finally Section VI concludes the paper.

II. SYSTEM ARCHITECTURE

A framework relies on a well-established measure for comparing the entities of two ontologies which are combined in a homogeneous way. The Figure 1 shows the system components.

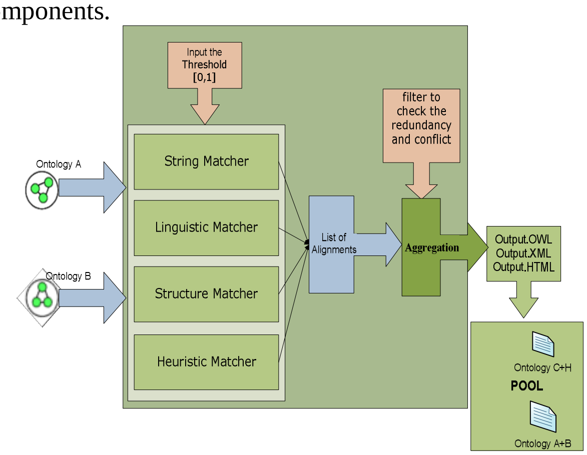


FIGURE 1: THE SYSTEM FRAMEWORK.

III. Detailed System

The system starts by loading two ontologies and extracts useful features such as class names, property names and subsumption relationships from them. In case ontology does not exist, we use our algorithm in [22] to transform relational database to OWL ontology.

A. String Matching

In general, the name of a class (i.e., label) is presented as a chain of characters without space characters. It is used to provide a human-readable description of class. Therefore, a name of class may be a word, or a combination of words. In fact, the name of each class should be unique in the ontology.

Terminological methods compare strings. Hence, these methods can be applied to the name, the label or the comments concerning entities to discover those which are similar. In general, it can be used for comparing class names and/or URIs.

A string matcher [2, 3, 7] usually takes as input the names of two concepts, then calculates the distance between them by distance functions that map a pair of strings to a real number. Consequently, the output will be a numeric value c $\in$[0, 1] to represent the confidence of the similarity. The main reason for using such measures is the fact that similar entities have similar names and descriptions across different ontologies.

String similarities are based on the assumption that the names of concepts and properties representing semantic similarity will have similar syntactic features. Thus, a string matcher usually first normalizes the input string of names and/or descriptions via stemming and tokenization. In the simplest form, the equality of tokens will be obtained and combined to give a score of the equality for the whole string.

In general, two properties are used to identify terms: the label and the name. The label is a string usually expressed in natural language to describe the purpose of the field to humans, while the name can be any string that is constrained by some name rules. These techniques are usually applied to names, labels, comments concerning entities and the URI. The scaled range is [0, 1] for comparing strings. To achieve high quality results and based on many experiments, the system disregards similarities that are smaller than a threshold of 0.65, and matches similarities greater than 0.65 to the full range [0, 1].

B. Linguistic Matching

The terminology used for naming and labeling concepts and properties is an important aspect of ontologies and provides information on the similarity between the ontology elements. However, linguistic features are also important for deriving an initial set of alignments to be refined by exploiting other kinds of matching. In fact, names of classes or properties are considered to provide one of the most important clues as to whether two terms are equal or not; therefore, we try to find relations between terms from different ontologies based on the details of their names. Such linguistic matching relies on algorithms and the use of external lexicon-based resources such as dictionaries, which are typically used to find close relationships such as synonymy between two terms and to compute the semantic distance between them in order to decide if a relationship holds.

This process is based on linguistic analysis [10,16]. There are two general techniques for label matching, the first of which employs linguistic analysis steps, such as abbreviations, avoiding recurrence and particle-ending. The other is matching the labels to determine the relationship between them.

In general, the linguistic similarity between terms is computed by considering labels and descriptions. Knowledge-based matchers take as input two concept (or synset) identifiers defined in WordNet [12] and produce semantic relations by exploiting their structural properties. They are often based on either similarity or relatedness measures. If the value of the measure exceeds the given threshold, a certain semantic relation is produced. Otherwise, “Idk” (I don’t know) is returned. This technique is implemented by using thesauri and WordNet, following an approach which is essentially the structural congruence between labels based on the hidden meanings of the words that they represent. WordNet, which takes two concept (synset) identifiers as input and returns the semantic relation holding between them, is considered not only to provide synonyms, hypernyms and hyponyms, but also to exploit additional structure to detect relationships between concepts (dinner→meal). For example, it considers synonyms as equivalent and hyponyms as subsumed, finding Match and Alignment to be similar classes (car→automobile).

In using a WordNet-based matcher we have to translate the (lexical) relations, which are provided by WordNet to logical relations [12], based on the following rules:

- A $\subseteq$ B, if A is a hyponym or meronym of B. For example, author is a hyponym of creator, therefore deducing that author $\subseteq$ creator.
- A $\supseteq$ B, if A is a hypernym or holonym of B. For example, Asia is a holonym of Jordan, therefore deducing that Asia $\supseteq$ Jordan.
- A = B, if A and B are connected by a synonymous relation or they belong to one synset. For example, quantity and amount are synonyms, therefore deducing that quantity = amount.
- A $\perp$ B, if A and B are connected by antonymy relations or are siblings in a part of hierarchy. For example, Jordan and Syria are siblings in the WordNet part of hierarchy, therefore deducing that Jordan $\perp$ Saudi Arabia.

C. Structure Matching

The aim of structural matching is to link an element of source taxonomy with an element of target taxonomy. The mappings generated are mainly specialization matches, based on calculations of the similarity of the source element with all those under the target taxonomy. Indeed, this kind of matching depends on what are considered the most important features of ontology nodes (e.g., class: super-classes and

Sub-class, property: super properties and sub properties). Therefore, similarity is based on the nodes of graphs.

The similarities between two concepts can be obtained from the language and from real attributes; and not only the similarities between the descriptions of their components, but also from similarities between the structures of the graphs representing them. The internal structure of similarities can be obtained by calculating the number of similar attributes divided by the attributes of a class.

Taxonomy is generally represented by an acyclic graph whose nodes are concepts and arcs corresponding to linked subclasses. Class inheritance analysis (is-a) considers the hierarchical connection between classes in order to identify “is-a” relationships.

Taxonomy (C, HC) includes a set of concepts C and a hierarchy subsumption between concepts HC. A concept is defined by its label and subclass relationships, which connect to other concepts. The label is a name (string) which describes entities in natural language and can be an expression composed of several words. Subclass relations establish links between concepts.

The intuition behind the algorithm is that if two concepts lie in similar positions with respect to is-a or part-of hierarchies relative to concepts already aligned in the two ontologies, then they are likely to be similar as well. For each pair of concepts (C1, C2) in the original list of alignment relationships, the structural matcher augments the original similarity value for pairs of concepts (C'1, C'2), such that C'1 and C'2 are equivalent to, are in an is-a relationship with or participate in a part-of relationship with C1 and C2 respectively. The augmentation depends on both the relationship and the distance between the concepts in the is-a and part-of hierarchies. It is important to mention here two important rules that help to ensure correct matching. First, if the super-concepts of two classes are the same, then these two concepts may be similar to each other. The second rule is that if the sub-concepts of two classes are the same, we can say that the concepts are also similar.

Structural analysis identifies identical classes by looking at their attributes and related (linked) classes. The main idea is that two classes of two ontologies are similar or identical if they have the same attributes and the same neighbor classes. Hence, matching concepts are based on structural similarity with regard to class hierarchy.

D. *Heuristic-based Strategies*

This phase of our system uses several features of ontologies (i.e., their structure, definitions of concepts and instances of classes) in order to find matches. Based on the idea that labeling is important and helps to align most of the entities, the structure also provides valuable help in identifying alignments. We have developed this step based on these two elements.

It considers the entities and structure of an ontology, i.e., class (C), property (P), relation (R), instance (I) and super-class (S). The distances between the input structures are then expressed in a set of equations. As described on Figures (2, 3).

$$Tot_{Sim(c,c')} = w * Sim_C(c,c') + \; w * Sim_P(c,c') + w * Sim_R(c,c') + w * Sim_I(c,c') + w * Sim_S(c,c')$$

where

- Sim$_C$ (c, c’) is the similarity between labels of classes,
- Sim$_P$ (c, c’) is the similarity between properties of classes,
- Sim$_R$ (c, c’) is the similarity between relations of classes,
- Sim$_I$ (c, c’) is the similarity between instances of classes,
- Sim$_S$ (c, c’) is the similarity between super-classes of classes,
- w is the weight, which is set at 1/5 in order to obtain results in the range [0,1],
- Tot$_{Sim(c,c')}$ is the average of all of these similarities, in the range [0,1]

FIGURE 2: HEURISTIC MATCHER EQUATION

The following is the function of heuristic match:

```
Function heuristicMatch (Ontology o1, Ontology o2) {
  for (All concept pairs (c, c’) where c ϵ o1 and c’ ϵ o2) {
$Sim_C$ = ComputeNameSimilarity (c, c’)
$Sim_P$ = (W*  findCommonAttributes (c, c’)) + (W*  matchDataTypes (c, c’))  + (W * matchDataInstance (c, c’))
$Sim_R$ = (W * findRelationship (c, c’)) + (W* matchNameRelationship(c, c’))
$Sim_I$= (W* findCommonInstance (c, c’) + (W*  matchInctance (c, c’))
$Sim_S$=W* ComputeNameSuperClass (c, c’)
    //compute overall similarity
  $Tot_{Sim(c,c')} = w * Sim_C(c,c') + \; w * Sim_P(c,c') + w * Sim_R(c,c') + w * Sim_I(c,c') + w * Sim_S(c,c')$
      }//end for
}//function heuristicMatch
```

FIGURE 3: HEURISTIC MATCHER FUNCTION

The output is one-to-one or one-to-many correspondences. This strategy is based on string similarity (i.e., Monge-Elkan metric [3]) structure and instances.

Monge-Elkan distance is recursive matching scheme for comparing two long strings s and t. By assuming that the strings s and t are broken into substrings (tokens), i.e., s = s1 . . . sK and t = t1 . . . tL. The intuition behind this measure is the assumption that si in s corresponds to a tj with which it has highest similarity. The similarity between s and t equals the mean of these maximum scores.

$$Monge - Elkan(A,B) = \frac{1}{|A|} * \sum_{i=1}^{|A|} \max_{J=1}^{|B|} match\,(A_i, B_j)$$

In heuristic matching, iteration is one of the most important steps in ontology alignment, which takes into account the structure of the input ontologies. It enables the whole process to be repeated several times, by propagating and updating the assessed similarities.

The sigmoid strategy combines multiple results using a sigmoid function, which is a smoothed threshold function, showing the importance of retaining high individual prediction values and removing low ones.

This technique starts after the application of the normalization process on the input elements, then compares class and property names in terms of editing distance and substring distance between entity names. The algorithms next create a distance matrix in order to determine the alignment group from the distance.

This algorithm is used in order to cover most possible features of ontologies (i.e., terminological, structural and extensional); on the other hand, we explicate recursive relationships and try to find the best matches through iteration. In general, this method faces problems when the viewpoints of two ontologies are highly different; thus, in order to achieve a high quality result, several of the above criteria should be combined, so that the rules which can be applied here are:

Any two concepts are probably the same if their labels are the same.

Any two concepts are equal if their properties are equal, even if their labels are different.

Two concepts that have the same instances are the same.

E. Aggregation

The results discussed here have been calculated using string, linguistic, structure and heuristic matchers. Indeed, with several matching strategies/ algorithms, there are several similarity values for a candidate matching (e1; e2). Therefore, combining them is an effective way to achieve high accuracy for a larger variety of ontologies, so this step extracts the combined matching result from the individual strategy results stored in the similarity cube. For each combination of ontology entities, the strategy-specific similarity values are aggregated into a combined similarity value, e.g., by taking the average or maximum value.

The easiest way to proceed consists of selecting correspondences above a particular threshold. Such threshold-based filtering allows us to retain only the most similar entity pairs. For the combination of the match results, the average value has been computed and a selection has been made using a threshold, for example: $\text{Semantic Distance}(\mathrm{C}, \mathrm{C}') \leq \text{Threshold}$

IV. Related Work

The following literature offers several approaches to the alignment of ontologies, based on measures of similarity.

A. The Naive Ontology Mapping

This approach [17] is simple, constituting a straightforward baseline for later comparisons. It comprises six steps. Feature Engineering demands that the ontologies be represented in RDF. Search Step Selection compares all entities of the first ontology with all entities of the second. Similarity Computation computes the similarity between entities in different ontologies, using a wide range of similarity functions. In Similarity Aggregation, NOM highlights individually significant similarities by weighting individual similarity results and aggregating them. This, however, neglects individual similarities that are of less significance. Interpretation uses the individual or aggregated similarity values to derive mappings between entities. Finally, Iteration repeats the previous step several times. This gives the capacity to access the already computed pairs and use more sophisticated structural similarity measures, whereas neglecting this step provides only a comparison based on labels and string similarity. A new version has more features and heuristic combinations, such as Quick Ontology Mapping (QOM) [18].

Advantage and Disadvantage: this approach applies string matching, structure matching and an instance matching, but it doesn't use auxiliary information. The means of defining the ontology is based on concepts, properties, and instances. The input-ontologies for this approach are in RDF format only. Moreover, it does not use a normalisation process. The way of selecting matching elements is threshold based.

B. PROMPT

Prompt [21] is a tool for merging ontologies, developed by Stanford University Knowledge Systems Laboratory. The knowledge model underlying PROMPT is frame-based and is compatible with Open Knowledge Base Connectivity. In general, this tool provides a semi-automatic approach to merging two ontologies; it is based initially on alignment relations, which should be held before providing output as a coherent ontology. More specifically, PROMPT performs some tasks automatically: it takes two ontologies as input and creates an initial list of matches based on class names. This list will be a coherent ontology. The following cycle then occurs: (1) the user triggers an operation by either selecting one of PROMPT's suggestions from the list or by using an ontology-editing environment to specify the desired operation directly; and (2) PROMPT performs the operation, automatically executes additional changes based on the type of the operation, generates a list of suggestions for the user, based on the structure of the ontology around the arguments of the last operation, and determines conflicts that the last operation introduced in the ontology, finding possible solutions for them. PROMPT then guides the user in performing other tasks for which his intervention is required. Its top level contains Classes (collections of objects arranged into hierarchies), Slots (binary relations), Facets (ternary relations) and Instances (individual members of classes).

Advantage and Disadvantage of PROMPT:

It applies string matching and semantic matching but it does not provide instance or structure matching. The input-ontologies for this approach are in different format like RDF(s), OWL-Lite, and OWL-DL. The output is merged ontology. The way of defining ontology is based on concepts, properties and instances. It does not deal with normalisation process. The way of selecting matching elements is based on highest value. This approach provides interactive suggestions to the users. It solves mismatches at terminological and scope of concept level, and it helps alignment by providing possible edit points and it supports repeatability. But it is not automatic which means every step requires user interaction.

C. *Chimaera*

Chimaera [19, 20] is a semi-automatic or interactive tool for merging ontologies. The engineer is in charge of making decisions that will affect the merging process. This tools starts by analysing the ontologies to be merged. It automatically finds linguistic match merges, and if it cannot find any matching terms, it gives the user control over any further action. In fact, it is similar to PROMPT, as both are embedded in ontology editing environments and offer the user interactive suggestions.

Advantage and Disadvantage of Chimaera:

It uses string matching, semantic matching and structure matching but it does not provide instance matching. The input-ontologies for this approach are OKBC ontologies and the output is a merged ontology. This approach analyses ontologies to be merged; if linguistic matches are found then the merge is processed automatically; otherwise, it uses subclass and super class relationship. In fact, this approach solves mismatches at the terminological level in a very light way, and provides interactive suggestions to the users. It solves mismatches at terminological and scope of concept level, and it helps alignment by providing possible edit points and it is not repeatability. But it is not automatic which means everything requires user interaction. (It is very similar to PROMPT).

V. Evaluation

It can be argued that the most significant and crucial issue when suggesting a new approach is its evaluation. Therefore, this section presents many test cases which are used to evaluate the performance of our system in different scenarios, followed by the experimental methodology, test data sets and results.

The Ontology Alignment Evaluation Initiative **(**OAEI) is a coordinated international initiative to establish agreement for evaluating and improving the available ontology alignment techniques. The OAEI ontology matching campaign is a contest organised annually since 2004, comprising several kinds of tests, processes and measures for assessing the results.

The benchmark data tests were divided into five groups, as shown in Table 1.

Table 1: Description of benchmark data set

Test Sets	**Ontology Description**	**Num of Ontologies**
101-104	Similar in both label description and hierarchy structure	4
201-210	Similar hierarchy structure	10
221-247	Similar in label description	18
248-266	Different in both label description and hierarchy structure	15
301-304	Real-world ontologies provided by different institutions	4

In order to assess the different approaches or evaluate the degree of compliance of the results of matching algorithms, standard information retrieval metrics are used, presenting four values which are widely used to estimate the quality of the alignment process and its results: precision, recall, overall and F-measure.

Currently, there are many ontology matching systems that have been developed based on different strategies for various purposes. In order to evaluate their performance and their qualities, we will focus on OAEI evaluation which employs a systematic approach to evaluate ontology matching algorithms and identify their strengths and weaknesses. After that we chose the following tests to show the evaluation:

A. *Tests 221 to 247*

In the third test set, the names, labels and comments had no special features that might confuse the alignment, but the structures of these ontologies were manipulated and some instances or/and properties were added. Therefore, in these ontologies our algorithm performed very well on string-, linguistic- and heuristic-based strategies in computing the similarity between features. This was due to the fact that the terms in these test cases had high string similarity; moreover, the heuristic matcher performed very well in these tests. On the other hand, where specific terms did not have similar names or comments, our algorithm was able to apply structural or semantic features of ontologies in order to derive the remaining alignments.

The most important issues affecting each of these are briefly stated here. Ontologies 221 to 247 featured no specialization (221), a flattened hierarchy test (222), an expanded hierarchy test (223), no instances (224), no restrictions (225), no datatypes (226), unit differences (227), no properties (228), class vs. instances (229) and flattened classes (230); all of these were matched with a very high recall and precision rate. As a conclusion, on this group of tests our algorithm performed well, which can be attributed to the fact that we carried out both syntactic and semantic similarity assessments.

Table 2: Result of tests 221-247

Test ID	**Precision**	**Recall**
221	1.00	1.00
222	1.00	1.00
223	1.00	1.00
224	1.00	1.00
225	1.00	1.00
228	1.00	1.00
230	1.00	1.00
231	1.00	1.00
232	1.00	1.00
233	1.00	1.00
236	1.00	1.00
237	1.00	1.00
238	1.00	1.00
239	1.00	0.99
240	1.00	0.99
241	1.00	1.00
246	1.00	1.00
247	1.00	1.00
Average	1.00	0.999

Although the structures of the candidate ontologies were changed, our algorithm found most correct alignments by using strings (label similarity, comment similarity), the linguistic perspective and heuristic matching. Therefore, both precision and recall were excellent.

While tests 221-247 shared the same names and comments, their structures differed. Instances were similar, but some ontologies did not contain them. The information given was sufficient to reach very good results. For most of these tests the structures were modified, which means that structural similarity was low, but the label similarity was very high. Because of this low structural similarity, the structure matcher did not work well for some tests; for example, tests 221, 232, 233 and 241 had high label and structural similarity factors, so both linguistic and structure-based strategies were employed, although the structure matcher made little contribution. Table 2 shows the results. Table 2 shows the results which appeared from tests 221-247.our results are very high and are nearly equal to 1. Our algorithms are heavily using linguistic and string matching algorithms.

B. *Comparison with other existing approaches*

In order to evaluate our system, a comparison of the system results was made against the published results from the 2007 Ontology Alignment Evaluation Initiative.

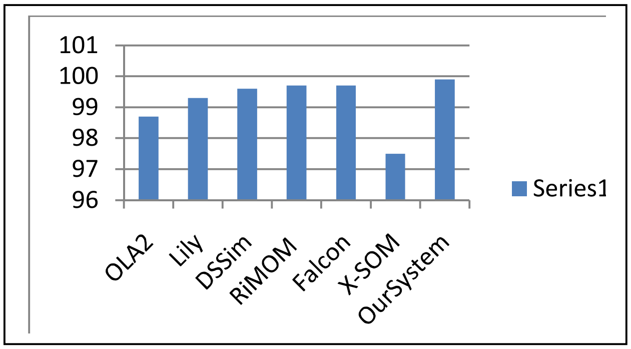


FIGURE 4: RESULTS OF TESTS 221-247

For most of tests 221-247, the structures of ontologies were manipulated, so that structural similarity was low; however, names, labels and comments in these ontologies had no special features, so linguistic similarity was very high. The information given was sufficient to yield very good results. In this set of tests, where the ontologies had high similarity with the reference ontology on linguistic information, our system performed very well and was the best, with precision, recall and F-measure scores of 1.00, 0.999 and 0.999 respectively. Other systems, including Falcon, DSSim and RiMOM also performed very well, with results on the F-measure of 0.997, 0.996 and 0.997 respectively.

VI. CONCLUSION

We develop new ontology alignment technique by using different matching strategies. This new ontology alignment approach utilizes both linguistic and structural information from ontologies in order to solve ontology alignment problems. The system is applying different matching algorithms, which includes: String matching, Linguistic-based strategies Structural matching, and Heuristic-based Strategies.

REFERENCES

[1] Berners-Lee T., Hendler J., and Lassila O., "The Semantic Web", Scientific Am, May 2001, pp. 34–43.

[2] Bunke H., Csirik J., "Parametric String Edit Distance and Its Application to Pattern Recognition", IEEE Transactions on Systems, Man, and Cybernetics**,** vol. 25, pp. 202-206, 1995.

[3] Cohen W.W., Ravikumar P., and Fienberg S.E., "A Comparison of String Distance Metrics for Name-Matching Tasks", In Proceedings of II Web, 2003, pp.73-78.

[4] Ehrig M., "Ontology Alignment: Bridging the Semantic Gap (Semantic Web and Beyond)", New York, Springer, 2006.

[5] Ehrig M., Euzenat J., "State of the Art on Ontology Alignment", Knowledge Web Deliverable D2.2.3, University of Karlsruhe, 2004.

[6] Euzenat J. , Shvaiko P., "Ontology Matching", Springer-Verlag, Heidelberg (DE), 2007.

[7] Euzenat J., Valtchev P., "Similarity-Based Ontology Alignment in OWL-Lite", In Proceedings of ECAI, 2004, pp.333-337.

[8] Euzenat J., Loup D., Touzani M., and Valtchev P., "Ontology Alignment with OLA", In 3rd EON Workshop, 3rd Int. Semantic Web Conference, 2004, pp. 333–337.

[9] Fensel D., "Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce", Springer, 2001.

[10] Giunchiglia F., Yatskevich M., "Element Level Semantic Matching", In Proceedings of the Meaning Coordination and Negotiation workshop at ISWC, (2004).

[11] Lambrix P. , Tan H., "A Tool for Evaluating Ontology Alignment Strategies", Presented at Journal Data Semantics, 2007, pp.182-202.

[12] Leacock C., Chodorow M., "Combining Local Context and Wordnet Similarity for Word Sense Identification", In WordNet: An Electronic Lexical Database, Christiane Fellbaum, MIT Press, 1998, pp. 265–283.

[13] Mao M., Peng Y., "The PRIOR+: Results for OAEI Campaign 2007", In Proceedings of OM, 2007.

[14] Nagy M., Vargas-Vera M., and Motta E., "DSSim - Managing Uncertainty on the Semantic Web", In Proceedings of OM, 2007.

[15] Taye M., " Ontology Alignment Mechanisms for Improving Web-based Searching", Ph.D. Thesis, De Montfort University, United Kingdom, England, 2009.

[16] Schorlemmer M., and Kalfoglou Y., "Progressive Ontology Alignment for Meaning Coordination: An Information-theoretic Foundation", In Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, Utrecht, The Netherlands, 2005, pp. 737–744.

[17] Ehrig M., Staab S., "Efficiency of Ontology Mapping Approaches", International Workshop on Semantic Intelligent Middleware for the Web and the Grid at ECAI 04, Valencia, Spain, August 2004.

[18] Ehrig M., and Staab S., "QOM - Quick Ontology Mapping", In Proceedings of International Semantic Web Conference, 2004, pp.683-697. .

[19] McGuinness D.L., Fikes R., Rice J., and Wilder S., "The Chimaera Ontology Environment", In Proceedings of the 17th National Conference on Artificial Intelligence (AAAI), 2000.

[20] McGuinness D.L., Fikes R., Rice J., and Wilder S., "An Environment for Merging and Testing Large Ontologies", In Proceedings of KR2000, 2000, pp. 483-493.

[21] Noy N.F., and Musen M.A., "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment", In Proceedings of AAAI/IAAI, 2000, pp.450-455.

[22] Alalwan N. ,Zedan H.,and Siewe F., "Generating OWL Ontology for Database Intgeration",In proceedings of Third International Conference on Advance in Semantic Processing 2009,pp.22-31.

Ontology Design Pattern Detection - Initial Method and Usage Scenarios

Muhammad Tahir Khan
School of Engineering, Jönköping University
P.O. Box 1026, SE-551 11 Jönköping, Sweden
Email: khmu09mt@student.hj.se

Eva Blomqvist
StLab, ISTC-CNR
via Nomentana 56, 00161 Roma, Italy
Email: eva.blomqvist@istc.cnr.it

***Abstract*—Ontology Design Patterns (ODPs) are emerging as an important support for ontology engineering. In this paper, we show how a method for detecting Content ODPs in existing ontologies can be used as a means to characterize online ontologies, e.g., for finding, browsing and analyzing them, as well as a means of analyzing an ontology being built, by detecting partial instantiations of a Content ODP in that ontology. The main contribution of this paper is the simple but effective method for pattern detection, together with its initial evaluation, as well as the study made on online ontologies providing an overview of Content ODP usage in real-world ontologies as well as a proof-of concept of the proposed method.**

***Keywords*-Ontology Design Patterns; Ontology Engineering;**

I. Introduction

Ontology Design Patterns (ODPs) are emerging as an important support for ontology engineering. On the semantic web, ontologies are no longer only constructed by developers having a background in logical languages and knowledge modeling. On the contrary, ontologies are commonly drafted by software engineers or found online, combined, and reused. Some small and frequently reused ontologies exist, e.g., the foaf ontology[1], however selecting and reusing larger and more complex ontologies is still a challenging task. Finding reusable ontologies is facilitated by ontology search engines, however, to understand and assess the ontologies is still up to the user, as well as formulating the keyword query to retrieve an accurate search result.

ODPs provide encoded best practices that can facilitate the construction of high-quality ontologies, despite lack of experience and deep knowledge of the logical languages (e.g., as experimentally shown in [2]). However, certain types of ODPs, e.g, Content ODPs, also come with a 'reference implementation', i.e., a small reusable component (usually represented in OWL [3]). A collection of such Content ODPs can be found in the *ODP Portal*[4], a wiki portal supporting the collection and management of ODPs.

In this paper, we show how a method for detecting such Content ODPs in existing ontologies can be used as a means to characterize online ontologies, e.g., for finding, browsing and analyzing them, as well as a means of analyzing an ontology being built, by detecting partial instantiations of a Content ODP in that ontology. The main contribution of this paper is the simple but effective method for pattern detection, together with its initial evaluation, as well as the study made on online ontologies providing both an overview of Content ODP usage in real-world ontologies as well as a proof-of-concept of the proposed method. In the following section, we describe Content ODPs in more detail. Section III describes related work, as well as two usage scenarios motivating our approach. In Section IV we present the pattern detection method, and its experimental validation is presented in Section V. Finally, in Section VI we conclude the paper and outline future work opportunities.

II. Content Ontology Design Patterns

There exist different types of ODPs having different characteristics, e.g., focusing on logical language constructs, architecture issues, naming, or efficient provision of reasoning services; for details on ODP types see [5], [6]. However, in this paper we focus on Content ODPs. Content ODPs are small ontologies with explicit documentation of design rationales, which can be used as building blocks in ontology design [5], [6]. As an example, we describe a Content ODP that is called *Agent Role*. It represents the relation between agents, e.g., persons, and the roles they play, e.g., professional roles such as researcher and teacher, as well as personal ones such as father and friend. Figure 1 shows an illustration of the OWL building-block representing this Content ODP. Content ODPs are collected and presented in different catalogues, such as the *ODP Portal*. In addition to their diagrammatic representation Content ODPs are described using a number of catalogue entry fields (c.f. software pattern templates), such as *name*, *intent*, *covered requirements*, *consequences*, and *building block* (linking to an OWL realization of the pattern). Reusing Content ODPs is a special case of ontology reuse, when the elements of the Content ODP are specialized, e.g., subclasses and subproperties that use domain-specific terminology are added, and more specific axioms are included.

III. Related Work and Motivation

The detection and analysis of *naming patterns* in ontologies was proposed in [7], where labels and other lexical entries are analyzed, e.g., for supporting refactoring. Although related in its aim, the approach uses lexical patterns to analyze the logical structure, while we use Content ODPs

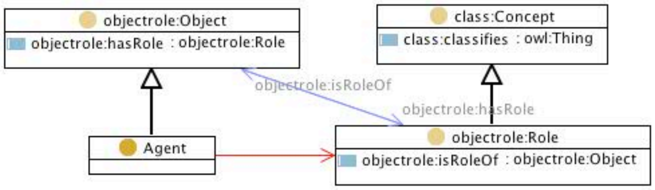

objectrole:Object
objectrole:hasRole : objectrole:Role
class:Concept
class:classifies : owl:Thing
objectrole:isRoleOf
objectrole:hasRole
Agent
objectrole:Role
objectrole:isRoleOf : objectrole:Object

Figure 1. The graphical representation, in UML, of the **Agent Role** ODP (the unlabeled arrow between Agent and Role representing disjointness).

as input. The approach in [8] for detecting *logical patterns* in ontologies using SPARQL queries is not applicable to our scenario since we are not only dealing with the logical constructs of the ontology, but the actual content, i.e., it is impossible to pose appropriate queries until the correct terminology has been established. In [9] a universal pattern language is introduced to monitor and detect constraint violations during ontology modeling, however, the approach is not focused on pattern detection in existing ontologies but rather detection of violations of the patterns being reused.

Our previous work includes OntoCase [5], a method for automatic ontology enrichment based on Content ODPs, mainly focused on enhancing ontologies generated from textual resources. The current method is a further development of the methods used in OntoCase. However, while OntoCase focused on finding matches that were 'useful enough' for interpreting the unrelated elements[1] of the input ontology, our focus in this paper is on simply finding instantiations of Content ODPs. OntoCase also applies an elaborate ranking scheme, which would be too computationally expensive to apply on a web scale, hence, its current implementation with focus on pattern ranking makes a direct comparison between OntoCase and the method proposed in this paper unrealistic.

Another approach that attempts to enrich ontologies by finding partial matches of Content ODPs and subsequently 'completing' the ontology by (automatically) adding the remaining part of the ODP as axioms in the ontology, is found in [11]. This approach does not assess if the missing parts are actually relevant and appropriate for that ontology, and where [11] chooses to add anything possible without evaluation, we leave such decisions up to the user by simply providing the pattern as a reference. Additionally, [11] uses a logical approach for the matching that involves both logical reasoning and manual pre-processing of the ODPs, hence it is not obvious that it will be feasible on a web scale.

A. Application Scenarios

Approaches exist for finding reusable ontologies, e.g., in the form of semantic web search engines such as Watson [12], Sindice [13], and SWOOGLE [14], and ontology repositories such as [15]. Recent improvements of search engines allow for simple visualization and assessment of the ontologies, e.g., by providing basic information on the size, language and complexity, as well as displaying key concepts [16]. Nevertheless, it is a difficult task to assess the usefulness of the ontology for a particular case. Moreover, merely to pose an appropriate keyword query to the system is challenging, since the index is based on the particular terminology of the ontology. Attempts have been made to introduce the 'query-by-example' paradigm into ontology search engines [17], however, in this case based on a user-developed model of the query and not considering specialization/generalization in the matching. In this paper, we propose to apply Content ODP detection for using ODPs as queries to find online ontologies that (partly) realize or specialize that Content ODP. This will address the need of users already knowing what kind of modelling issue they are interested in, wanting to find online ontologies containing particular solutions for that modelling problem. One could imagine ontology repositories that are browsable by means of the different ODPs the ontologies contain. In addition, it provides an interesting possibility to study the type of modelling solutions applied in online ontologies, as well as to study the 'support' that Content ODPs have in online ontologies (see Section V-C1).

A second scenario is concerned with novice users building an ontology 'from scratch'. Studies such as [2] show that ODP selection is a problem that hampers the designers in fully exploiting the benefits of ODPs, hence, additional tool support should be provided for Content ODP selection. Recently some support has appeared, in the form of the XD Tools [18] for the NeOn Toolkit [19]. XD Tools currently provide ODP registry browsing and search facilities for retrieving Content ODPs based on a keyword query. The semantic vector search service extends standard keyword indexing, but does not take into account the fact that ODPs are usually more abstract than the terminology in the ontology. In this scenario we have a draft ontology and a set of Content ODPs, and wish to find some patterns that are already partially realized within the ontology. Hence, we try to detect occurrences of each pattern in the draft ontology, and if such occurrences are found we propose the pattern to the user, who can study the pattern and evaluate his or her solution against the best-practices that the pattern describe. Although other approaches have been proposed, to use patterns to enhance draft ontologies, they have either been purely automatic (e.g., the OntoCase approach [5]) or using a heavy logical approach also requiring the manual pre-processing of the ODPs, as in [11]. We propose the use of a light-weight method that will easily detect simple modelling attempts of a novice user, who receives a list of possible patterns of interest to be used as a means for evaluating or enhancing the ontology by specializing the Content ODPs found (see Section V-C2).

[1] In this paper the term *ontology element* (formally defined in [10]) denotes formal expressions used to represent any entity, e.g. named classes and properties are elements, as well as class definitions and restrictions.

IV. Content ODP Detection

The method for Content ODP detection is based on OntoCase [5]. The basic principle of OntoCase pattern detection is to identify specializations of Content ODPs based on matching the terminology used for properties and classes, as well as matching the property structure through domain and range restrictions.The graph based pattern matching of OntoCase is currently not considered in this paper because of its complexity, e.g., processing time. The approach presented in this paper is thereby both an extension and a simplification of the OntoCase approach. The main extension is that the current approach not only matches domain and range restrictions but all axioms of the ontology, while a simplification is that the current approach uses less computationally expensive algorithms, e.g., the OntoCase ranking has been removed, instead relying on simple matching percentages.

The proposed approach uses three main methods for detection; (1) import detection, (2) direct matching, and (2) indirect matching. Import detection is the trivial detection of an explicit import of a pattern URI in the chain of the ontology's import closure. We are aware of the fact that imported patterns might not be used in the ontology, but taking this into account is still future work. The direct matching aims to detect clones of the pattern, including partial clones, existing in the ontology, while the indirect matching aims to detect specializations of a pattern in the ontology, i.e., where the pattern classes and properties have been exchanged for more (domain-) specific ones. The procedure is described using pseudocode in Figure 2.

Require: A pattern p, an ontology o, and the thresholds of class/property/axiom matches t_1, t_2, t_3.
Ensure: The matching percentages P_c, P_p, P_a or $null$ if match was below threshold.

```
if import closure of $o$ contains an owl:import of $p$ then
    return $P_c = 100, P_p = 100, P_a = 100$
else
    $oClassNames = getClassNames(o)$
    $\ldots$
    $extend(oClassNames, oPropertyNames)$
    $extend(pClassNames, pPropertyNames)$
    $classMatches, propMatches, axiomMatches = \emptyset$
    for each set in $pClassNames$ = $pElement$ do
        $pairs = stringMatch(pElement, oClassNames)$
        if $pairs \neq \emptyset$ then
            $classMatches = classMatches + pairs$
    $P_c = percentage(classMatches, p)$
    for each set in $pPropertyNames$ = $pElement$ do
        $pairs = stringMatch(pElement, oPropertyNames)$
        if $pairs \neq \emptyset$ then
            $propMatches = propMatches + pairs$
    $P_p = percentage(propMatches, p)$
    for each axiom of $p$ = $a$ do
        $pairs = tripleMatch(a,$ axioms of $o)$
        if $pairs \neq \emptyset$ then
            $axiomMatches = axiomMatches + pairs$
    $P_a = percentage(axiomMatches, p)$
    if $(P_c > t_c)\&(P_p > t_p)\&(P_a > t_a)$ then
        return $P_c, P_p, P_a$
    else
        return $null$
```

Figure 2. Detection procedure.

If no import was found the detection procedure starts by retrieving all the terms to be used, e.g., local names (i.e., excluding the namespaces) and labels of classes and properties. The *extend()*-function uses heuristics to extend the terms into term sets, each representing one original element, e.g., class or property, of the ontology or the pattern respectively. The intuition is that we need to allow for certain variations in the matches, i.e., for the direct matching this would include heuristics for capitalization and morphological variations, while for the indirect matching this includes also term specialization, for instance, using background knowledge such as WordNet [20]. The *stringMatch()*-function performs exact string matching on the elements of the pairs $A \times B$, where A is one of the extended term sets from the pattern and B is the set of all such extended term sets of the ontology. Depending on the previous application of heuristics, this may return several matching pattern element-ontology element pairs. The axioms are matched based on matching the subject-predicate-object structure of their contained triples, i.e., the *tripleMatch()*-function above. It converts all the axiom constituents to sets of strings, similar to above but taking into account the uniqueness of the OWL constructs, e.g., 'reserved words' such as `disjointWith` does not need to be extended with synonyms and lexical variations, then matches the string sets for subject, predicate, and object respectively. The *percentage()*-function calculates the fraction of classes/properties/axioms that are in the matched set, with respect to the overall number of the pattern or ontology. If above the threshold values, the match is confirmed. If needed, the details of each match can also be recorded, although not shown in the procedure above.

V. Experimental Validation

To allow for a proof-of concept validation of the approach it has been implemented as a stand-alone Java application, and applied to two different datasets.

A. Implementation

The method described in Section IV has been implemented using Java. The implementation exploits the OWL API (3.0) [21] for handling the ontologies, the Watson API [22] for retrieving online ontologies, and JAWS [23] for interfacing WordNet and supporting the indirect matching. In this implementation the extension heuristics for the direct matching are restricted to (i) ignoring capitalization, and (ii) recognizing the most common ways of replacing spaces in element names, e.g., using the camel convention or _ instead of spaces. For the indirect matching these heuristics are extended by using JAWS. Through a simple lookup mechanism the corresponding synset of every element name or label in the pattern is retrieved from WordNet, and all specializations (hyponyms) of that synset are additionally added to the extended term set. No disambiguation of the terms are currently performed, i.e., all possible chains of synsets are used, but since only specialization and not generalization is considered this is a manageable set. In the current implementation only direct matching is performed on the axiom triples, i.e., no specialization of these are allowed.

B. Data Collection

Two sets of ontologies were collected, each corresponding to one of the usage scenarios described previously.

1) Online Ontologies: The online ontologies were retrieved through the keyword search feature of the Watson API [12]. The set of keywords entered are matched to the local names, labels, comments, or literals of elements occurring in semantic documents, i.e., ontologies. Based on query logs of the Watson search engine we collected a list of the 70 most used search keywords. From this list we further selected the 50 keywords that returned the highest number of ontologies, in order to get a sample that represents typical search results. A list of matching ontologies was retrieved, for each of the keywords. The results were filtered based on language (i.e., only allowing RDF/OWL). Next, all broken links were filtered out, e.g., where the ontology was no longer accessible at that URI. The resulting set consisted of 845 ontologies, which were saved locally for repeatability reasons[24] (no additional sampling was performed).

2) Ontology Drafts: The ontology drafts result from student assignments[2] to design an ontology within the theater domain, based on a fixed set of requirements, i.e., competency questions (CQs) [26]. They had not previously been introduced to ODPs (assured by self-assessment) but had some training (one full day) on OWL and ontology engineering. The task was designed so that it would be possible to solve some of the design problems using Content ODPs, in order to expose the students to those problems before introducing Content ODPs later in the course. The students were given 3 hours to solve the exercise, and they all used the same tool. The resulting set consists of 15 ontologies (of between 9-20 classes, and 12-30 properties).

C. Accuracy of Implicit Content ODP Detection

To evaluate the Content ODP detection implementation we have applied it to the two sets of ontologies, together with a set of 76 content ODPs (the complete set of Content ODPs at that time available from the ODP Portal).

1) Online Ontologies: The accuracy evaluation of the indirect ODP detection within online ontologies was for practical reasons performed on a small sample of ontologies, randomly selected from the data set. The sample contained 40 ontologies, where 33 of them had at least one match to any of the 76 Content ODPs (threshold of class matches set to 50%), summing up to a total of 200 pattern detections. The ontologies were then reviewed by a human evaluator, to assess the matches. The human evaluator classified each proposed detection (i.e., each pattern-ontology pair) either as (a) 'I agree that there is a match, the suggestion is correct', (b) 'I do not agree that there is a match, the suggestion is incorrect', or (c) 'I cannot decide based on the available information'. The evaluator classified 62% of the suggestions into category (a), i.e., correct matches, 31% into (b), i.e., incorrect, and 7% into (c). Counting (a) as the correctly suggested patterns, **62%** corresponds to the level of **precision** of the detection approach.

While 62% precision may seem low, it is comparable to other complex search mechanisms operating on online data that are currently widely appreciated. For instance, consider online search engines, where the precision on complex queries has been assessed in [27]. The average precision of the search engines in this study ranged from 51.25% for Google, down to 32.5% for Ask.com, for complex queries. Although this study has a completely different aim, it shows that in fields such as online information retrieval, a precision as low as 51.25% is considered acceptable.

2) Ontology Drafts: The task given to the students had a clear set of requirements (CQs) and was constructed with a set of Content ODPs in mind, i.e., 6 of the Content ODPs available in the portal, hence, also the recall of the approach could be assessed. To obtain the 'gold standard' on which to perform the recall calculation, we additionally analyzed the intents and requirements of all the 76 Content ODPs and recognized 13 additional Content ODPs where the requirements match the CQs. Table I shows the resulting set of 19 Content ODPs [3]. Most of these were compositions or generalizations of the smaller set, while a few also represented alternative modelling choices applicable in this context, e.g., to view a music album as a collection of tracks, or tracks as being proper parts of the album.

Table I
CONTENT ODPS APPROPRIATE FOR USE IN SOLVING THE TASK.

	Content ODP Name		Content ODP Name
1.	Agent Role	11.	Person
2.	Collection/Collection Entity	12.	Place/Location
3.	Componency	13.	Region
4.	Co-participation	14.	Situation
5.	Information Realization	15.	Time-indexed Participation
6.	N-ary Participation	16.	Time-indexed Part Of
7.	Object Role	17.	Time-indexed Person Role
8.	Participant Role	18.	Time-indexed Situation
9.	Participation	19.	Time Interval
10.	Part Of		

The results of the indirect matching can be seen in Table II. The table shows the average precision and recall over the set of 15 ontologies (threshold for class matches again at

[2] Assignment details at [25]

[3] Collection and Collection Entity are represented by the same OWL-building block although having separate pages in the ODP portal, just as Place and Location. Since our method works on the OWL building blocks, they are here treated as the same pattern.

50%). The reader should however note that while precision is highly relevant, recall is not an entirely relevant measure from a user perspective since some patterns are overlapping or simply specializations of others. This means that even with a recall less than 100% the set of proposed ODPs could cover the complete task.

Table II
AVERAGE PRECISION AND RECALL OVER THE 15 ONTOLOGIES.

Method	Avg. Precision	Avg. Recall
Direct matching only	14.4%	1.4%
Direct+Indirect matching	66.4%	38.9%

As a comparison we note that using only direct matching, on average 1 pattern was proposed for each ontology, while using the indirect matching the system proposed on average 11 patterns for each ontology, which is considered a reasonable number to be assessed by the ontology engineer (compared to the catalogue of 76, hence, 86% of the patterns were filtered out). An interesting problem to consider is how the method would perform on larger ontologies, however, it is worth noting that many patterns are applicable in several places within an ontology (e.g., if ‘partOf’ is applicable, it will only be proposed once although applicable throughout the ontology). This indicates that the number of proposed patterns will not explode when the ontology size increases.

Comparing to the XD Tools search functionality, when entering the ontology requirements, i.e., the CQs that were the basis of the draft ontologies, into the search interface (standard keyword indexing) of XD Tools only reaches a precision of 36% and recall of 21% for the first 11 results (i.e., the average number of patterns suggested by our pattern detection method). When considering the first 20 results we also note that the precision drops to 20% while the recall remains on 21%, indicating that we do not get any more useful results even if we check the next ten results, i.e., some patterns are very hard to retrieve with standard keyword-search.

D. Content ODPs Detected in Online Ontologies

The study of online ontologies additionally aimed to assess Content ODP usage. In Table III we present the count of ontologies where the 20 most frequent patterns were detected using our method as described above (class match threshold again set to 50%). The dataset consisted of 682 ontologies previously collected[4].

When analyzing these results we note that certain patterns are favored by the background knowledge used for the matching, e.g., the Constituency pattern contains only one concept, named ‘entity’, which appears at the top level of WordNet. Other patterns are instead never matched, due to the simple property matching applied, e.g., the Part Of pattern, which contains no classes at all but only two properties. This leads us to conclude that the numbers presented are probably not reliable as an absolute count, rather the result can be seen as an indication that the solutions proposed by patterns are in fact used in online ontologies. More precise matching needs to be applied in order to derive accurate statistics. However, through our experience in working with patterns we can confirm that the patterns in Table III are in fact a selection of the ones we, as ontology engineers, have used most frequently, although some very frequent ones could not be detected due to limitations in the heuristics.

Table III
THE 20 MOST FREQUENTLY DETECTED CONTENT ODPS AND THE NUMBER OF ONTOLOGIES WHERE THEY WERE DETECTED.

Content ODP Name	#	Content ODP Name	#
Constituency	666	Topic	68
Participation	204	Classification	67
Componency	148	Description	67
Co-participation	101	Parameter	67
Types of Entities	101	Basic Plan Execution	59
Collection	98	Participant Role	47
Agent Role	85	Task Role	44
Region	75	Task Execution	44
Object Role	73	N-ary Participation	37
Communities	70	Situation	34

VI. CONCLUSIONS AND FUTURE WORK

We have presented a simple Content ODP detection method and described its proof-of-concept implementation and evaluation in two usage scenarios. Although the accuracy of the current implementation can certainly be improved, we believe that it is a valuable complement to traditional ontology search and retrieval, where keyword-based search in most cases returns few or no results when applied to the task of finding highly abstract Content ODPs. In addition, we have presented a small study on Content ODP support in real-world ontologie. The results show that many Content ODPs are widely used, although very few seem to be explicitly imported.

Future work includes to improve the heuristics used by the method, especially on the side of properties and axioms where the results at the moment are quite poor. Typical naming patterns for properties could be used in order to compare property names with different structure but a similar meaning. Additionally, the axiom matching could be extended to exploit the matching results already provided by the class and property matching, in order to achieve an indirect axiom matching method as well. OntoCase, as it is currently implemented, is not applicable to our usage scenarios, however as future work it would also be interesting to compare other possible relaxations of OntoCase, to improve the trade-off between scalability and accuracy. We are currently exploring the possibility to

[4] Unfortunately, 162 of the original 845 ontologies were not processable through the OWL API, due to syntactic errors or other problems.

integrate the implementation as a selection service in the XD Tools plugin within the NeOn Toolkit, and in the future we will also consider the possibility of providing a detection service for online search, to bring the advantages of this approach into practice.

ACKNOWLEDGMENTS

Part of this work was performed with support from The Swedish Foundation for International Cooperation in Research and Higher Education, project DEON (grant #IG2008-2011).

REFERENCES

[1] [Online]. Available: http://xmlns.com/foaf/0.1/ Accessed: 08.10.2010

[2] E. Blomqvist, A. Gangemi, and V. Presutti, "Experiments on Pattern-Based Ontology Design," in *K-CAP 2009*. ACM, 2009, pp. 41–48.

[3] [Online]. Available: http://www.w3.org/2002/07/owl Accessed: 08.10.2010

[4] [Online]. Available: http://www.ontologydesignpatterns.org Accessed: 08.10.2010

[5] E. Blomqvist, "Semi-automatic Ontology Construction based on Patterns," Ph.D. dissertation, Linköping University, Department of Computer and Information Science at the Institute of Technology, 2009.

[6] A. Gangemi and V. Presutti, "Ontology Design Patterns," in *Handbook on Ontologies, 2nd Ed.*, ser. International Handbooks on Information Systems. Springer, 2009, pp. 221–243.

[7] O. Sváb-Zamazal and V. Svátek, "Analysing Ontological Structures through Name Pattern Tracking," in *Proceedings of EKAW 2008*, ser. Lecture Notes in Computer Science, A. Gangemi and J. Euzenat, Eds., vol. 5268. Springer, 2008.

[8] O. Sváb-Zamazal, F. Scharffe, and V. Svátek, "Preliminary Results of Logical Ontology Pattern Detection using SPARQL and Lexical Heuristics," in *Proceedings of WOP 2009, collocated with ISWC-2009*, vol. 516. Washington D.C., USA: CEUR Workshop Proceedings, October 2009.

[9] O. Noppens and T. Liebig, "Ontology Patterns and Beyond - Towards a Universal Pattern Language," in *Proceedings of WOP2009 collocated with ISWC2009*, vol. 516. CEUR-WS.org, November 2009.

[10] [Online]. Available: http://ontologydesignpatterns.org/cpont /codo/codolight.owl Accessed: 08.10.2010

[11] N. Nikitina, S. Rudolph, and S. Blohm, "Refining Ontologies by Pattern-Based Completion," in *Proceedings of WOP 2009, collocated with ISWC-2009*, vol. 516. Washington D.C., USA: CEUR Workshop Proceedings, October 2009.

[12] M. d'Aquin, C. Baldassarre, L. Gridinoc, M. Sabou, S. Angeletou, and E. Motta., "Watson: Supporting next generation semantic web applications." in *Proceedings of the WWW/Internet conference, Vila real, Spain, 2007*, 2007, pp. 363–371.

[13] G. Tummarello, E. Oren, and R. Delbru, "Sindice.com: Weaving the Open Linked Data," in *Proceedings of ISWC/ASWC2007, Busan, South Korea*, ser. LNCS, vol. 4825. Berlin, Heidelberg: Springer Verlag, November 2007, pp. 547–560.

[14] L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. C. Doshi, and J. Sachs, "Swoogle A Semantic Web Search and Metadata Engine," in *Proc. 13th ACM Conf. on Information and Knowledge Management*, Nov. 2004.

[15] N. F. Noy, N. H. Shah, B. Dai, M. Dorf, N. Griffith, C. Jonquet, M. J. Montegut, D. L. Rubin, C. Youn, and M. A. Musen, "BioPortal: A Web Repository for Biomedical Ontologies and Data Resources," in *Poster and Demo Proceedings of ISWC 2008*, 2008.

[16] S. Peroni, E. Motta, and M. d'Aquin, "Identifying Key Concepts in an Ontology, through the Integration of Cognitive Principles with Statistical and Topological Measures," in *Proceedings of ASWC 2008*, ser. Lecture Notes in Computer Science, J. Domingue and C. Anutariya, Eds., vol. 5367. Springer, 2008, pp. 242–256.

[17] C. Anutariya, R. Ungrangsi, and V. Wuwongse, "SQORE: A framework for semantic query based ontology retrieval." in *Advances in Databases: Concepts, Systems and Applications*, ser. LNCS, vol. 4443. Springer, 2007, pp. 924–929.

[18] [Online]. Available: http://stlab.istc.cnr.it/stlab/XDTools Accessed: 08.10.2010

[19] V. Presutti, E. Daga, A. Gangemi, and E. Blomqvist, "eXtreme Design with Content Ontology Design Patterns," in *Proc. of WOP 2009, collocated with ISWC-2009*, vol. 516. Washington D.C., USA: CEUR Workshop Proceedings, October 2009.

[20] C. Fellbaum, Ed., *WordNet - An Electronic Lexical Database*. MIT Press, 1998.

[21] [Online]. Available: http://owlapi.sourceforge.net/ Accessed: 08.10.2010

[22] [Online]. Available: http://watson.kmi.open.ac.uk/WS_and_ API.html Accessed: 08.10.2010

[23] [Online]. Available: http://lyle.smu.edu/∼tspell/jaws/index.html Accessed: 08.10.2010

[24] [Online]. Available: http://ontologydesignpatterns.org/experi ments/ODPDetectionCorp2010.zip Accessed: 08.10.2010

[25] [Online]. Available: http://ontologydesignpatterns.org/wiki /Training:PhD_Course_on_Computational_Ontologies_%40_ University_of_Bologna/Hands-on_(Day2):_Theater _productions Accessed: 08.10.2010

[26] M. Gruninger and M. S. Fox, "The role of competency questions in enterprise engineering," in *Proc. of the IFIP WG5.7 Workshop on Benchmarking - Theory and Practice*, 1994.

[27] M. L. Robinson and J. Wusteman, "Putting Google Scholar to the test : A preliminary study," *Program*, vol. 41, no. 1, pp. 71–80, 2007.

Towards Dynamic Ontology

Integrating Tools of Evolution and Versionning in Ontology

Perrine Pittet, Christophe Cruz, Christophe Nicolle
LE2I, UMR CNRS 5158
University of Bourgogne - Dijon, France
{perrine.pittet, christophe.cruz, christophe.nicolle}@u-bourgogne.fr

***Abstract*—Since Gruber's definition, a lot of works focused on evolution or versioning issues. Not much attention has been paid to integrated solutions which resolve both these two purposes. In this paper we present a new semantic architecture that combines versioning tools with the evolution process. This architecture called VersionGraph is integrated in the source ontology since its creation in order to make it possible to evolve and to be versioned.**

Keywords-evolution; versioning; Versiongraph; ontology lifecycle;change operations;

I. INTRODUCTION

Many works have been published about the definition of ontology to bridge the gap of semantic heterogeneity. Literature now generally agrees on the Gruber's terms to define an ontology: explicit specification of a shared conceptualization of a domain [1]. The domain is the world that the ontology describes. It can be a general domain or a more specific one. This description uses a vocabulary of concepts which is understandable and agreed by people of the domain; here is the meaning of "shared conceptualization". The ontology can be implemented in several languages with a different level of formalization and expressivity, with no ambiguity that's why ontology is an "explicit specification". The development of ontology is becoming a common task and an inescapable supportfor information systems interoperability [2]. This research domain is mature and the first feedbacks arise. New scientific deadlocks are identified concerning the lifecycle of ontology especially the evolution phase. Discussing about those issues leads us to first ask what part of the ontology definition is concerned by this lifecycle and where the evolution can be situed. Regarding to [3] Ontology lifecycle depends from changes occurring in the domain, conceptualization and/or specification of the ontology. Moreover, as depicted in Figure 1 (red dotted arrows), a change on one of this identified sources can impact a change in the other sources. Figure 1 shows the causes of changes related to the domain (a), the conceptualization (b) and the specification (c). We can notice that a change cause in (a) and (b) can have a change consequence in (b) and (c). Proposition a new classification of the identified changes in the state of art of [26] we have identified two types of change interaction:

Firstly, the domain can impact conceptualization. These changes are similar to changes in database schemas [5]. For example new concepts/relationships must be considered or existing concepts/relationships must be improved or deleted. That's the role of Domain Evolution [6] or Domain Fusion (Ontology Integration [7], Ontology Merging [8]) proposals. The domain can also affectspecification. For example a complete translation to a new specification corresponds to Ontology Translation_1[9] proposals.

Secondly, the conceptualization can impact the specification. For instance new models in the domain are introduced and require a change in the concept/relationships organization, formalization and expressivity. That's the purpose of Conceptualization Evolution [10] and Conceptual Revision (Ontology Debugging [11]) proposals. Nevertheless, we note that four types of change, used to resolve conceptual heterogeneity (conceptualization part), don't impact the ontology itself: Ontology Mapping [12], Ontology Matching [13], Ontology Articulation [13] and Ontology Morphism [14]. These last ones add an external mapping to bridge the semantic gap. We argue that a change in the specification doesn't impact the conceptualization or the domain when the specification language is enough rich to express this change. It's the case of Description Logics languages[15] whichdisplay different levels of expressivity by holding different ontology constructors.So, we can choose one of them depending on the level of expressivity we need.

From this discussion, we deduce that the evolution phase concerns the domain and the conceptualization of the ontology. The Conceptualization Evolution is a direct consequence of the Domain Evolution. The new research area which aims at resolving the impact of change management on ontology is known as Ontology Dynamics [16] Ontology Dynamics deals with all issues concerning changes impacting the ontology (change of the domain, change in the conceptualization, or change in the specification), especially maintenance and evolution. The ontology development is a dynamic and incremental process starting with the creation of a brute ontology which has to be revised, refined and populated [17]. In the literature, a lot of papers have addressed the problem of managing the lifecycle of the existing ontology [18]. Most of them propose tools dealing with the different causes of change as depicted in figure 1. The major part put the emphasis on the evolutionissues [19]. Some articles cope with versioning solutions to handle different versions of evolved ontologies [20]. Nevertheless not much attention has been paid to the characterization of an ontology which integrates in its definition the mechanisms to evolve and being versioned. We can cite the proposition of [40,41], which approach is quite similar to ours but differs in its final solution.

This paper focuses on a generic architecture make it possible to combine the definition of ontology withevolution and versioning operators. This architecture can be used with any type of ontology based on description logics and especially OWL-DL formalism [21]. An implementation of this architecture is presented at the end of this paper. It is an extension of the Jena's library [22] by override ofthe existing ontology handling operators.

This paper is articulated in three parts. The first part presents a background on ontology evolution and versioning. The second part describes the VersionGraph Architecture. The last part is an example of evolution versioning based on the Wine Ontology [23].

II. EVOLUTION AND VERSIONNING BACKGROUND

According to [24], ontology lifecycle is divided in seven steps: needs detection, conception, management and planning, evolution, diffusion, use, and evaluation.The needs detection phase starts witha detailed inventory of the domain and the various purposes.Like evolution phase, conception phase needs: knowledge acquisition, shared conceptualization building, formalization (Semantic Web formalisms[25]…) and integration of the existing resources (other ontology, applications…).The phase of management and planning underlines the importance of having a constant monitoring and a global policy to detect or initiate, prepare or evaluate the lifecycle iterations. This work intends to guarantee that an iteration of the lifecycle is activatedwhen an evolution is ready to be completed. The management step requires tools not only to prepare the ontology to adapt the domain changes but also to keep trace of the previous versions of the ontology. These goals can be reached with a versioning system [26].Diffusion phase deals with the deployment of the ontology. The use phase encloses all the activities related to the access of the ontology. Finally, the evaluation phase aims at evaluating the ontology state. Moreover, like the needs detection phase,it collects beforehand the knowledge of the domain and can also rely on previous studies or feedbacks. Except for the evolution and management phases, all the steps described can be considered as mature domains. Furthermore, this description of the lifecycle shows that evolution and managementremains the most complex phases. Evolution is the backbone of the lifecycle iterations. Therefore, the change management process is totally based on it.

The rest of our state of art is articulated in three parts. According to the literature, we will first define the evolution role, operations and process. Then we'll have a look at the existing solutions for change representation and ontology versioning. We will see how to link the evolution process and a versioning system in order to integrate both of them in existing ontologies.

A. *Ontology Evolution*

As stated by [26], ontology evolution aims at responding to one or several changes in the domain or the conceptualization by applying them on the source ontology.This brief definition looks abstract and leads us to ask: what kind of changes does the evolution apply? How evolution applies them? What are the criteria to respect? How can we manage a goodevolution? Evolution changes are defined in the literature and especially in [9] as a succession of simple or complex operations the user wants to apply on the intension (schema) or the extension(data) of the ontology.This evolution aims at adapting the ontology to the changed domain. Applying and propagating thechange are often manual tasks but can be done automatically by synchronization with the domain.According to [27] these tasks usually occur during the use phase of the ontology. Ontology Dynamics clearly define the evolution criteria. [28] and [29] qualify the maintenance of the ontology as the most important criterion.Evolution has to maintain whatever relies on the ontology.Maintaining the ontology consistent and pertinent, in a consensus is an inescapable issue of evolution[30]. Applying changes on ontology can turn the conceptualization inconsistent and irrelevant. That's why an evolution should never be validated before the user has a preview of the impact of the changes on the ontology. This impact can only be estimated if the evolution operations are semantically clearly defined.

In order to assure that this process is fully respected, some works propose an approach in six phases. 1. the **change detection** phase consists in detecting what changes occurred in the domain or in the point of view must be propagated to the conceptualization. Lots of papers in the Ontology Dynamics deal with this phase and propose methods and tools like integrated event handlers[27], ontology learning [31] etc… 2. the **representation phase** aims at representing the selected changes with ontological operations. [10] classifies the evolution operations in two types: elementary (atomic) operations and composed (complex) operations.According to [10], elementary operations are simple operations that modify only one entity like addition/suppression of classes/relations, of hierarchy, domain, range links, of class/relation properties like disjoint, transitivity, etc…whereas composed operations are a composition of several elementary operations.The choice of composed operations depends on the granularity of the evolution needs. Therefore, we aimat displaying our proposition to the major part of formal ontologies. So we need to integrate usual operations. Usual operations correspond to operations the ontology that developers are the most expected to use when creating and evolving an ontology. In addition to elementary operations, the literature gives some lists of usual operations (e.g. [32,33]). In complement, we have extracted other usual operations like "change the place of an entity"…from the application Protégé. Moreover we make a distinction between operations on the intension and operations on the extension. The cited works on change operations don't specify specific operations for the instances because they argue that an instance can become a class [10]. However, we maintain that schema operations can't be confounded with instance operations. Actually, it is impossible to create an instance (instance operation)related to a class if this class is not created. Inversely a class can be created (schema operation) without instances. 3. the **semantic phase** prevents the user from inconsistency risks by determining the sense of the represented changes. For example, if composed operations have been selected, this phase will allow seeing their decomposition in elementary operations. 4. the **implementation** of the changes alerts the user of the impact on data in terms of data gain or loss. [10] gives these impacts from a list of 22 usual operations (the elementary ones and some composed). 5. the **propagation phase** aims at informing all the dependent parts of the ontology (other ontologies, application) of these changes. 6. Finally, comes the **validation** of the changes. In the following part we will see how our proposition can integrate these operations in the versioning system and follow these evolution phases.

B. Versioning

This section is articulated in three parts. First we define the role of versioning, bringing our new vision on this definition; Then we describe the versioning process of our versioning system based on the 6 phases of evolution process. A state of art on the existing solutions of change representations will help us to build the tools needed in this process. Finally we present our suggestion to permit the identification and the retrieval of a version of an ontology. [26] gives in 2007 a very strict definition of the role of versioning : give a transparent access to different existing versions of an ontologyby creating a versioning system.This system identifies the versions by their “Id” and delimits their mutual compatibility. In the past three years, Ontology Dynamics proposals extend its role: manage several chronologic and multitemporal versions [34], at local or web level [35], when collected, distributed, accessed by search engines [35]. All these definitions correspond to a retroactive versioning because versions of the ontology have to preexist. However in our objective, we want to integrate a versioning system since the creation of the first version of the ontology. Therefore, we need, as the ontology development, a dynamic and incremental process, which could take into account a new version at each evolution phase. That’s why we propose to merge the evolution process (following the 6 phases) with the versioning one.

First, the user chooses the list of operations to apply (cf. change detection phase). The versioning system formalizes them (cf. representation phase), turn them semantically understandable (cf. semantic phase), records and implements them (cf. implementation phase).Then after the propagation of the changes, (cf. propagation phase), the user validates them (cf. validation phase) and the versioning system applies them and generates the new version of the ontology corresponding to an evolution iteration. Finally the versioning system can give a transparent access to both of the versions with criteria defined by the user [36]. It can delimit compatibility by retracing evolution operations [32, 33]. To follow this process, we need to specify the tools displayed by our versioning system. According to [37], a change specification should enclose an operational change specification (our list of operations), then the conceptual relationship between the first version and the new one (the selected operations on the selected entities).The first phase of the evolution process is then completed. The next step is to represent these changes.Several approaches are proposed in the literature to represent changes. Major part of them uses logs.Versioning logs [38] record the different versions of an ontology by representing each entity at a given time. For each class, relation and instance, a new instance of “EvolutionConcept” class is created. [37] argues that metadata should be added to identify this change. In versioning logs, each instance is annotated with metadata (Id, cause, transaction time, state validated or not…).This solution is interesting if the versioning log can be integrated in the ontology. However for our purposesthere is no need to represent each entity if it’s not modified by the evolution. Evolution logs [39] don’t save the versions but act like a change history. Not each entity but each substitution in the ontology is recorded in order to be reused when the user wants to access a version.Tracing the substitution rather corresponds to our objectives as a substitution contains the selected operations and the entities affected. In order to cope with our evolution process we propose to create a Version concept like in the versioning logs integrated in the ontology that will be created at each evolution iteration. This Version concept encloses: 1/the substitutions operated in the intension or 2/ those operated on the extension and3/ the metadata.Then, the implementation phase can be helped by introducing event detectors on data. In the application Jena supporting the ontology, the idea is to insert methods using “ActionListener” objects. The propagation phase can be performed by generating events activating the “ActionListener” objects. Finally, the validation is similar to the “Commit” operator of a DBMS, can be done by a simple click by the user. Our incremental versioning process following the 6 evolution phases constitutes the first part of our versioning system.

The second part corresponds to the transparent access definition. The first issue is the identification of the versions. Most of the versioning systems use “Id” of theontologies to identify them [35]. Though, it’snot enough to identify in which version a change on a certain entity occurred. As we have introduced metadata and the list of substitutions occurred when a Version is created, those data can serve as search criteria to identify and retrieve the right version. We have chosen to extend Jena operators (access on ontology etc…) in order to take into account the search criteria. This extension can be performed by an override of the access methods. For example, by adding metadata and operation attributes. This state of art permitted us to build the evolution and versioning process of our proposition. We also managed to design the versioning tools in order to represent changes and access the ontology.

III. VersionGraph Architecture

In this section, we present the VersionGraph architecture which implements the choices of our state of art. First, we focuses on the operations corresponding to the evolution operations. Then we describe our versioning system. Finally, we give an example of evolution on the Wine ontology.

A. *Evolution Operations*

Contrarily to the [4] proposition, the schema and instance operations are differentiated respectively by `SchemaOperation` and `InstanceOperation`. `SchemaOperation`type operations correspond to the creation and deletion of classes (`AddClass`) and properties (`AddProperty`) but also to additions and deletions of restrictions on them. We distinguish restrictions on the classes and properties or properties of the data link hierarchy (`HierarchyLink`) such as class / subclass, property / sub-property. Also in the class restrictions, limitations like classes / properties such as the relationship between properties and classes (`ClassPropertyLink`, `ClassDataPropertyLink`), cardinality (`ClassPropertyCardinality`) are classified. Also in the restrictions we find domain and range restrictions of attributes (`PropertyAttributeLink`). Finally `TypeProperty` operations are used to define a specific constraint of a property (transitive, symmetric etc ...). `InstanceOperation`type operations, correspond to operations of addition and deletion of individuals and statements about these individuals. We distinguish between the assertions relying individuals to the values (`DataPropertyAssertion`) and those specifying the types for these individuals (`ObjectPropertyAssertion`).

B. *From evolution to versioning*

From these evolution operations and the study of the different versioning solutions of our state of art, we derived a versioning system. At each evolution of the ontology, the system stores in the ontology, the changes impacted by the operations used and the context. This versioning system is an independent ontology which intends to be integrated into the existing ontology by a simple addition operation. Then, the user can start a first evolution of ontology in choosing whether to change the schema (intension) or data (extension) using the above operations. Each list of changes chosen by the user during the evolution is kept using a concept `SchemaVersionGraph` for `SchemaOperation` operations and `InstanceVersionGraph`for `InstanceOperation`operations on instances by specifying which elements of the ontology are concerned (concepts, relationships...). Contextual information can be added (as version, date, author, description...). These data are traced during the evolution using a concept of context `VersionContext`. The set containing `SchemaVersionGraph` or `InstanceversionGraph` and `VersionContext`is called `VersionGraph`. Figure 2 depicts an overview of the ontology schema. For more clarity, it only shows concepts and their relationships under 6th hierarchical degrees. In a transparent way, each application of changes made by the user generates a new `VersionGraph`.The`VersionGraph` definition in Protégé is presented in Figure 3.As depicted in this figure a `VersionGraph` contains a link with the previous version of the ontology (`hasPreviousVersionGraph`). It's actually a link to the core ontology (for the first `VersionGraph`) or to the previous `VersionGraph`.Because of its nature, our system of evolution and versioning can be integrated into applications using ontologies Jena. The access operations of the library Jena can be overridden by the criteria of change and context. Until now, proposals for versioning are often accompanied by a specific application that the user must install to access the version it wants if the use of URI is not enough (Evolva). However, many ontologies are accessed using a Java API Jena. Indeed, this library supports ontology-based formalisms like RDF, RDFS, OWL and the various DAML + OIL. Jena contains all the methods to access and edit ontologies. In addition, it also implements all the basic operations of evolution and the commonly used composed ones. Overridden access methods are able to take into account the criteria of versions thanks to new attributes. These criteria are integrated into the ontology itself as we saw in the previous paragraph.

C. *The Wine Ontology Versionning*

The Wine ontology is an ontology example in which international wines are described. For the first step, we import the VersionGraph ontology into the Wine ontology by an addition operation. Then the system creates the first version of the wine ontology with a first instance of `VersionGraph`. This Versiongraph only has a link with the source ontology.

```
<vg :VersionGraph#VersionGraph0>
        p:hasPreviousVersionGraph
        <http://www.w3.org/TR/owl-guide/wine.rdf>;
```

Then we want to add the “StrawWine” wine which doesn’t exists in the Wine ontology. Straw Wine’s fruit is selected then dried in the sun so that the juice is very concentrated in flavor and sugar. So it is a dessert style wine sometimes heavy or balanced or straw gold color. It can be made from red grapes Cabernet Franc and Cabernet Sauvignon or Chardonnay white grapes and Sauvignon Blanc. To add this new concept and describe it, the system creates another `VersionGraph`. This new one islinked with the previous one.The system specifies a SchemaVersionGraph which contains the operations needed to describe and add the concept in the ontology.

```
# VersionGraph1 description
<vg:VersionGraph#VersionGraph1>
```

```
        p:hasPreviousVersionGraph<vg:VersionGraph#V
ersionGraph0>;
        p:hasDate "11/05/2010";
        p:hasAuthor  "Perrine PITTET";
        p:hasSchemaVersionGraph
<vg:SchemaVersionGraph#SchemaVersionGraph1>;
# AssociatedSchemaVersionGraph1 description
<vg:SchemaVersionGraph#SchemaVersionGraph1>
        p:hasAddClass  <rdfs:class#StrawWine>;
        p:hasAddClassHierarchyLink
<vg:ClassHierarchyLink#ClassHierarchyLink1>;
        p:hasAddClassDataPropertyLink
<vg:ClassDataPropertyLink#ClassDataPropertyLink1>;
        p:hasAddClassDataPropertyCardinality
<vg:ClassDataPropertyCardinality#ClassDataProperty
Cardinality1>;
        p:hasAddClassDataPropertyCardinality
<vg:ClassDataPropertyCardinality#ClassDataProperty
Cardinality2>;
# Description des SchemaOperation utilisées
<vg:ClassHierarchyLink#ClassHierarchyLink1>
        p:class <rdfs:class#StrawWine>;
        p:subClass <rdfs:subClassOf#DessertWine>;
<vg:ClassDataPropertyLink#ClassDataPropertyLink1>
        p:class <rdfs:class#StrawWine>;
        p:dataProperty <owl:DataProperty#hasColor>;
        p:value <rdf:resource#Golden>;
<vg:ClassDataPropertyCardinality#ClassDataProperty
Cardinality1>
        p:class <rdfs:class#StrawWine>
        p:dataProperty <owl:DataProperty#hasBody>
        p:value      <rdf:resource#Full>      and
<rdf:resource#Moderate>
<vg:ClassDataPropertyCardinality#ClassDataProperty
Cardinality2>
        p:class <rdfs:class#StrawWine>
        p:dataProperty
<owl:DataProperty#madeFromGrape>
        p:value   (<rdf:resource#CabernetSauvignon>
and       <rdf:resource#Carbernetfranc>)       or
(<rdf:resource#Chardonnay>                    and
<rdf:resource#SauvignonBlanc>)
```

Then, we want to add an individual of Straw Wine type: "Vin Paillé de Corrèze". First, we need to validate the previous changes by a "Commit". Then changes in the schema are recorded and the new schema version is propagated to the ontology. A third `VersionGraph` is generated for the addition of the individual. This time it contains an `InstanceVersionGraph`.

```
# VersionGraph2 description
<vg:VersionGraph#VersionGraph2>
        p:hasPreviousVersionGraph
<vg:VersionGraph#VersionGraph1>;
        p:hasDate             "12/05/2010";
        p:hasAuthor     "Perrine PITTET";
        p:hasInstanceVersionGraph
<vg:InstanceVersionGraph#InstanceVersionGraph1>;
#AssociatedInstanceVersionGraph1 description
<vg:InstanceVersionGraph#InstanceVersionGraph1>
        p:hasAddIndividual <vg:AddIndividual#AddInd
ividual1>
        p:hasAddMemberClass <vg:AddMemberClass#AddM
emberClass1>
        p:hasAddObjectPropertyAssertion
<vg:AddObjectPropertyAssertion#AddObjectPropertyAssertion1>
# InstanceOperationdescription
<vg:AddIndividual#AddIndividual1>
        p:individual <rdf:resource#VinPaillé>
<vg:AddMemberClass#AddMemberClass1>
        p:individual <rdf:resource#VinPaillé>
        p:class <rdfs:class#StrawWine>
<vg:AddObjectPropertyAssertion#AddObjectPropertyAssertion1>
        p:individual <rdf:resource#VinPaillé>
        p:objectProperty <owl:ObjectProperty#locatedIn>
        p:value <rdf:resource#FrenchRegion>
```

IV. CONCLUSION

Ontology evolution and versioning are recent domains of search. Most of current ontology versioning approaches are not based on the evolution process. Rare are the solutions which integrate these mechanisms since the creation of the ontology. Our proposed architecture Versiongraph is a semantic solution towards the characterization of a dynamic ontology which reaches these objectives. Our ongoing research shows preliminary results on evolution of several ontologies like Wine, FOAF or Pizza. Our short coming plan is to enhance our evolution and versioning process on several projects applied to online press comments, tourism and town heritage ontologies.

REFERENCES

[1] Gruber, T.,R. - A translation approach to portable ontologies- Knowledge Acquisition ,1993.

[2] Moguillansky, &al - A Theoretical Model to Handle Ontology Debugging & Change Through Argumentation – Proc. of the IWOD at ESWC 2008, Karlsruhe, Germany. 2008.

[3] Klein, M., Fensel, D. - Ontology Versioning on the Semantic We- . s.l. : SWWS Standford, 2001, SWWS Stanford.

[4] Jaziri W., A methodology for ontology evolution and versioning, The Third International Conference on Advances in Semantic Processing (SEMAPRO™2009), pages 15-21, ISBN: 978-1-4244-5044-2, October 11-16, 2009, Sliema, Malta.

[5] Ventrone, V., Heiler, S. - Semantic Heterogeneity as a Result of Domain Evaluation. 1991, SIGMOD Record Special Issue: Semantic issues in Multidatabase Systems.

[6] Klein, M., Noy, N. - A Component-Based Framework for Ontology Evolution., F. 2003. IJCAI-03 Workshop on Ontologies and Distributed Systems, CEUR-WS, vol. 71.

[7] Calvanese, D., &al -A Framework for Ontology Integration - in Cruz, I., Decker, 2002.

[8] Pinto, H.S., &al - Some Issues on Ontology Integration – Proc. of the Workshop on Ontologies and Problem-Solving Methods (KRR5) at 16th International Joint Conference on Artificial Intelligence (IJCAI-99).

[9] Avesani, P., &al. - A Large Scale Taxonomy Mapping Evaluation - s.l. : Lecture Notes in Computer Science, Springer, 2005, Vol. Volume 3729. 67-81.

[10] Noy, N. F., Klein, M. - Ontology Evolution: Not the Same as Schema Evolution -Stanford Medical Informatics, Stanford University, Stanford, CA, USA Vrije University Amsterdam, Amsterdam, The Netherlands.

[11] Haase, P. & Qi, G. 2007. - An Analysis of Approaches to Resolving Inconsistencies in DL-based Ontologies – Proc. of the IWOD at ISWC 2007, pp. 97-109.

[12] Kalfoglou, Y., Schorlemmer, M. - Ontology Mapping: the State of the Art- Knowledge Engineering Review,18 (1), pp. 1-31 2003.

[13] Hu, W. & Qu, Y. - Block Matching for Ontologies- Proc. of the 5th International Semantic Web Conference (ISWC-06), pp. 300-313.

[14] Flouris, G. & Plexousakis, D. - Handling Ontology Change: Survey & Proposal for a Future Research Direction-Technical Report FORTH-ICS/TR-362. 2005.

[15] Baader, F., &al - The Description Logic Handbook : Theory, Implementation & Applications - Cambridge University Press, pp. 495-505, 2003.

[16] Ontology Dynamics : http://www.ontologydynamics.org/od/

[17] Djedidi, R., Aufaure, M., A. - Change Management Patternsfor Ontology Evolution Process – Proc. of the IWOD at ISWC 2008, Karlsruhe, Germany. 2008.

[18] Ribeiro, M., &al.- Belief Contraction in Web-Ontology Languages – Proc. of the IWOD at ISWC 2008, Karlsruhe, Germany. 2008.

[19] Pan, J., Z. - A Stratification-based Approach for Inconsistency Handling in Description Logics, Proc. of the IWOD at ISWC 2007, Innsbruck, Austria. 2007.

[20] Allocca, C., &al - Detecting Different Versions of Ontologies in Large Ontology Repositories – Proc. of the IWOD , Karlsruhe, Germany. 2008.

[21] OWL : http://www.w3.org/TR/owl-features/

[22] Jena: http://jena.sourceforge.net/

[23] Wine Ontology : http://www.w3.org/TR/owl-guide/wine.rdf

[24] Hodgson, R.- The Potential of Semantic Technologies for e-government- presentation of eGov Open Source Conference-Washington, DC, March 18th, 2003

[25] Semantic Web: http://semanticweb.org/wiki/Main_Page

[26] Flouris, F., &al - Ontology Change: Classification & Survey - The Knowledge Engineering Review, 1–29, 2007, Cambridge University Press

[27] Tovar, E., Vidal, M., E. - REACTIVE: A Rule-based Framework to Process Reactivity – Proc. of the IWOD at ISWC 2008, Karlsruhe, Germany. 2008.

[28] Atle Gulla, J., Sugumaran, V. - An Ontology Creation Methodology: A Phased Approach.. Karlsruhe, Germany : s.n., 2008. Proc. of the IWOD at ISWC 2008.

[29] Dividino, R. and Sonntag, D. - Controlled Ontology Evolution through Semiotic-based Ontology Evaluation. Karlsruhe, Germany : s.n., 2008. Proc. of the IWOD at ISWC 2008.

[30] Zablith, F., &al - Using Background Knowledge for Ontology Evolution. 2008, Proc. of the IWOD, Karlsruhe, Germany.

[31] Novacek, V., &al - Semi-automatic Integration of Learned Ontologies into a Collaborative Framework.

[32] Stojanovic, L., &al - User-driven Ontology Evolution Management. 13th Int. Conf. on Knowledge Engineering and Knowledge Management. 2002.

[33] Stuckenschmidt, H., Klein, M. - Integrity and Change in Modular Ontologies. 18th International Conference on Artificial Intelligence, 2003.

[34] Grandi, F. - Multi-temporal RDF Ontology Versioning. Karlsruhe, Germany, IWOD at ISWC 2008.

[35] Allocca, C., &al - Detecting Different Versions of Ontologies in Large Ontology Repositories.

[36] Stuckenschmidt, H., Klein, M. - Integrity and Change in Modular Ontologies, 18th Int. Joint Conference on Artificial Intelligence, 2003.

[37] Klein, M., Fensel, D. - Ontology Versioning on the Semantic Web. SWWS Standford, 2001

[38] Yildiz, B. - Ontology Versioning and Evolution, Asgaard, 2006

[39] Liang, Y. - Ontology Versioning and Evolution For Semantic Web-Based Applications. 2005.

[40] Sassi, N., &al - From Temporal Databases to Ontology Versioning: An Approach for Ontology Evolution, In Ontology Theory, Management and Design: Advanced Tools and Models, Ed IGI-Global Publisher, USA, 2010.

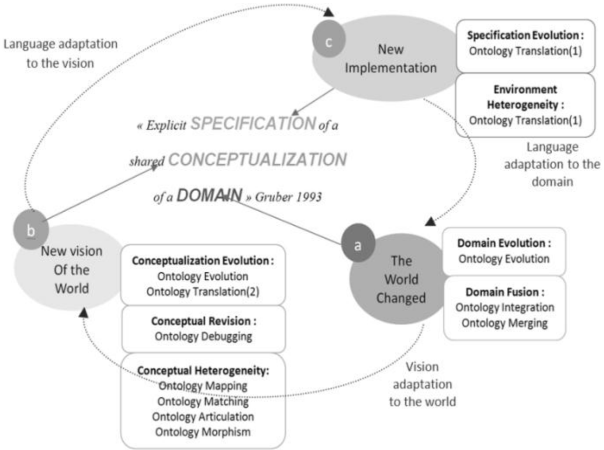
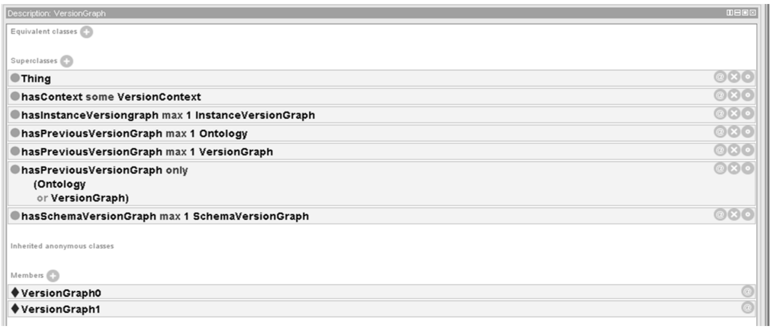


Figure 1. Causes of changes in the lifecycle of an ontology.


Description: VersionGraph

Equivalent classes

Superclasses

- Thing
- hasContext some VersionContext
- hasInstanceVersiongraph max 1 InstanceVersionGraph
- hasPreviousVersionGraph max 1 Ontology
- hasPreviousVersionGraph max 1 VersionGraph
- hasPreviousVersionGraph only
  (Ontology
  or VersionGraph)
- hasSchemaVersionGraph max 1 SchemaVersionGraph

Inherited anonymous classes

Members

♦ VersionGraph0
♦ VersionGraph1

Figure 3. VersionGraph definition in Protege.

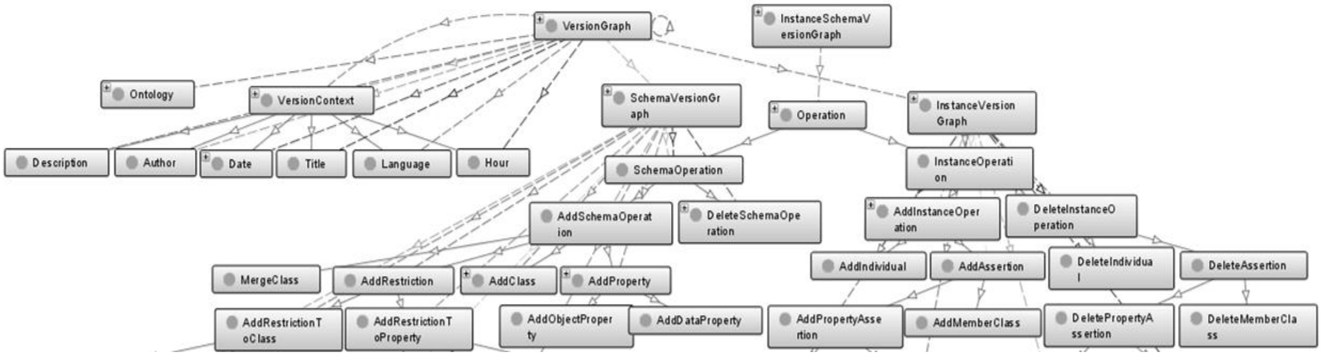


Figure 2. Overview of the VersionGraph Ontology

Identifying Beginner Problems in Expressing Domain Semantics when Developing Ontologies

Sandra Lovrenčić, Mirko Maleković
University of Zagreb
Faculty of organization and informatics
HR-42000 Varaždin, Croatia
sandra.lovrencic@foi.hr, mirko.malekovic@foi.hr

Abstract—**Ontologies as a knowledge representation method are already being applied in various areas. Therefore, this method is introduced to new developers constantly. The paper investigates possibility for overlooking ontology features that can enable users to properly represent semantics of the domain of interest. In initial research, ontology development using frames was considered and evaluation was made based on criteria connected to classes, hierarchy and attributes. Possible beginner oversights are identified. Suggestions considering chosen semantic criteria are also described.**

Keywords-knowledge representation; ontology development; domain semantics; semantic criteria

I. Introduction

Knowledge representation with the help of ontologies is a subject of research for two decades already. With new technologies emerging every day and Semantic Web vision, they have become an important part of various research areas, including knowledge management. Along with their features, their application is also spreading.

In a well known paper [1] ontology is defined as “an explicit specification of a conceptualization” and frame systems are described as knowledge representation framework for “describing hierarchies of classes with slots” that ontologies consist of. Over years, many ontology development methods, languages and tools have been evolved [2]. Ontology evaluation is, of course, an integrating part of their development and was in centre of research about five years ago. Evaluation of ontology content is concentrated on consistency, completeness, conciseness, expandability and sensitivity, whereas ontology taxonomy evaluation considers inconsistency, incompleteness and redundancy [3][4]. Well known OntoClean method evaluates ontologies according to rigidity, identity, unity and dependency, concepts introduced from philosophy [5][6]. Ontology evaluation can be based on structural, functional and usability-profiling measures [7] as well as on “coverage of a particular domain and the richness, complexity and granularity of that coverage; the specific use cases, scenarios, requirements, applications, and data sources it was developed to address”[8]. Factors considered in the evaluation process can be features of languages and tools used [9], but also user demands and simplicity of use [10]. Evaluation methods can be combined to explore various ontology characteristics [11].

Some aforementioned evaluation methods are designed to be conducted independently of ontology development methods, tools or languages used and others consider them as possible biases that can influence on richness of knowledge representation. However, a factor of knowledge and experience of ontology engineer is rarely taken into consideration. To new developers ontologies are constantly introduced as a knowledge representation method. If accustomed to different means of representing knowledge, such as classical databases, they may not use all features that ontologies offer for representing the semantics of the domain of interest, for example, description of classes with the use of instances of other class. Defining and focusing on potential oversights when teaching or learning how to develop ontologies can reduce initial mistakes. Therefore, the main goal of presented research was to evaluate basic ontology elements, such as classes, hierarchy and attributes (slots) with the purpose to discover how well beginners can understand and exploit the concept of ontologies when managing and representing knowledge.

The paper is organized as follows: in second section semantic criteria for ontology evaluation are introduced; afterwards, research process as well as analysis and results are described; conclusion and future work are in the final section.

II. Semantic Criteria

When considering the use of ontologies, “the most important aspect of the ontological representation is its capacity of expressing domain semantics”[12]. Generally, ontologies represent semantic knowledge of a certain domain through hierarchies of classes and attributes (and their constraints) that describe them. Therefore, those features should be used for a proper domain description, but some of them may be overlooked, especially with the lack of experience. Detection of those oversights can give valuable information about important parts of ontology development lessons.

At the Faculty of organization and informatics in Varaždin, Croatia, ontologies are taught at two levels:

- second or third year undergraduate students learn ontology development at simple level within Knowledge Management course, where only frame systems as knowledge representation formalism are introduced and students have no prerequisites that include formal logical systems;

- second year graduate students learn ontology development with OWL and description logic reasoning within Knowledge Bases and Semantic Web course and have prerequisites that include formal logical systems.

For initial research with the purpose of simplicity and further guidelines, only first case is considered. Since the goal was to discover how easy beginners can grasp the concept of ontology and how many semantics will they be able to represent with it, ontology elements used within frames - classes, hierarchy and attributes (slots) were obvious choice for analysis.

During previous years, it was noted that students who develop ontologies for the first time tend to describe classes only with simple string or integer attributes and that they are inclined to either develop very poor hierarchies with many attributes or very rich ones where even instances are mixed for classes. For that reason lectures were organized in a manner that each covered one specific development part: detailed description of development process with examples, development of classes and hierarchy and the use of attributes. The hypothesis was that beginners will to some extent overlook the use of more complex ontology features – complex attributes, use of more hierarchies and their connection for better domain description. According to important ontology elements, several criteria were taken into consideration for the evaluation:

Total number of hierarchies – Although this is not commonly, for the purpose of research, class hierarchy was divided into two parts: main hierarchy (describing the domain of interest) and support hierarchy (used to better describe the domain of interest). For example, University studies ontology has several such hierarchies: types of studies, teaching participants, courses, conduction places and enrolment requirements [13]. Because the domain of interest was types of studies, this would be main hierarchy. Other hierarchies would help in its description – their classes or instances would be used as values for class attributes in main hierarchy. Therefore, this criterion can imply more semantically versatile description.

Number of support hierarchies – There can be several main hierarchies in complex domains, as well as support ones. Because support hierarchies are those that designate more complexity in domain description, it is necessary to determine their actual number (if any).

Depth of main and support (where applicable) hierarchies – It is obvious that hierarchies with more branches and more depth give better description of domain structure and class relations and therefore represent a valuable criteria. Main hierarchy can be the only hierarchy in one-hierarchy ontology or one or more of those that directly describe the domain of interest in multiple-hierarchy ontology. These criteria are considered with hypothesis that support classes will have lesser depth than main ones.

Total number of classes, number of classes in main and support hierarchies (where applicable) – The number of support hierarchies and hierarchy depth cannot itself give complete information about the degree of semantics represented: main hierarchy should obviously have a number of classes, but support hierarchy can actually consist of only one, whose instances must be values for a certain attribute. Only ontologies that have support hierarchies were evaluated according to these criteria, whereas all ontologies were used for analysis of total number of classes.

Total number of attributes – It is needless to say that attributes are the real descriptors of classes and that their greater number should mean better semantic representation. For this criterion the total number of attributes is taken, regardless whether they belong to main or support hierarchy.

Number of attributes in main and support (where applicable) hierarchies – These criteria gives even better insight in how well is which part of ontology described. Of course, it is applicable only on ontologies that have support hierarchies.

Number of connecting attributes – Connecting attributes are those that connect classes together, primarily meaning that the attribute value of one class is the instance of the other (regardless whether it is a part of main or support hierarchy). They show how well are represented connections among various parts of the domain, and the actual effect of support classes – how much semantic they add.

Number of simple and complex attributes – The last two criteria show the complexity of class description. Attributes are divided into two groups, simple and complex. Simple attributes are any, boolean, float, integer and string whilst complex are class, instance and symbol.

It should be noted that those criteria are chosen according to main ontology elements using frames. They can be proven more or less useful after the research and need for other criteria can be discovered.

III. Research

Research was conducted at Faculty of organization and informatics during spring semester of year 2009/2010.

A. Participants

As already described, students are taught knowledge representation with ontologies during laboratory exercises in course Knowledge Management. For that reason, research was conducted with second and third year undergraduate students at this course (year of course enrollment is not fixed; only prerequisites are). Participants had no prior experience with ontologies, but were familiar with knowledge representation methods for knowledge management in general. Total number of students was 152 in 10 groups. Ontology development is part of their final grade, but several irregular students decided to apply for the regular exam and not to present their work.

Laboratory exercises were divided into two parts. First part is not the subject of this research, but was good introductory for ontologies: students had to collect knowledge about some topic in knowledge management domain, represent it in a wiki system and tag important concepts that were then visualized in graph.

Assignments from both parts of laboratory exercises were included in student grades with 25% in total. Also, students had to obtain at least 12 of 25% to be able to apply even for a regular exam. This ensured their motivation to accomplish given tasks.

B. Research Process

As a tool for ontology development was chosen Protégé [14], as one of most used open-source tools that has good user interface and support and is being developed for more than 20 years [15]. Version that was used is Frames without Protégé Axiom Language (subset of first order logic axioms), for several reasons:

- participants were undergraduate students with no prerequisites that included knowledge of first order logic;
- although they had mostly the same courses in their first year, students can choose between two directions in their undergraduate studies, information systems and business systems – therefore, their interest and knowledge of informatics topics is not the same;
- Protégé editor is very intuitive and allows easy manipulation with ontology elements of interest for the research.

Because of grading, each student had to choose a different domain for ontology, according to hers/his interests. Domains could be similar, but not exactly the same (for example, car models from two different manufacturers). Their task was to represent the chosen domain with ontology as best as possible and to incorporate into it all features that were taught to them.

Laboratory exercises consisted of four sessions. Activities at each session are described below:

Session 1 – Students were taught about ontologies through example of University studies ontology [13]. Firstly, the role of classes and their attributes in hierarchy was explained to them. Then they had a task to create a small hierarchy example. Protégé-Frames tool was also presented to them with step by step explanation how to create ontology. Their next task was to try out the tool. Students also had enough time to start searching for a suitable domain according to their preferences and interests. They had to find a domain for ontology development until next session. As already explained, they had to have different ontologies.

Session 2 – The most important task for this session was to create one or more class hierarchies. Each student's hierarchy was individually controlled and they were given suggestions for better arrangement of classes. Also, at least one support class for better semantic description of the specific domains was proposed to each of them.

Session 3 – For this session the most important task was to create appropriate attributes and connect with them all hierarchies together (where applicable). Suggestions for more use of complex attributes and explanations how to use attributes to connect different classes were also given to each student.

Session 4 – The last session was actually used for presentation and grading of ontologies. Students had to finish ontologies at home (create frames for instance entry window, populate ontology with enough instances to be able to make queries, create several queries, visualize ontology). About half of students already created some attributes at second session and populated instances at third, so they had enough time for the completion of the task.

IV. Analysis and Results

142 students delivered their ontology. As mentioned before, one of goals was also to determine whether some changes in semantic criteria definition have to be done. Therefore, for initial analysis 50 randomly selected ontologies were used. With purpose of better understanding of research results, information about values that have small range (0-4) is presented in Table 1. Following statistical measures were used for semantic criteria analysis: arithmetic mean, median, mode, standard deviation and skewness.

TABLE I. Selected Criteria Values

Criteria	Values				
	0	*1*	*2*	*3*	*4*
N. of hierarchies	-	30	14	1	5
N. of main hierarchies	-	49	1	0	0
N. of support hierarchies	30	14	2	4	0
Depth of main hierarchy	2	5	21	16	6
Depth of support hierarchy	14	4	2	0	0

The raw data from Table 1 already shows that less than a half of ontologies have support hierarchies and that almost all of them have only one main hierarchy. The support hierarchies generally have 0 depth (one class), and the depth of main ones is satisfactory. First two rows for value 0 are empty, because all ontologies have at least one main hierarchy. Detailed analysis is given in next subsections.

A. Classification Analysis

Table 2 shows average values obtained for selected ontologies according to following criteria: total number of hierarchies, number of support hierarchies, depth of main and support hierarchies and number of classes in total, and in main and support classes, where applicable.

It can be seen that most ontologies had only one hierarchy (median and mode are 1). Actually, 20 of 50 (40%) had at least one support hierarchy, meaning that more than half of students did not use this ontology feature to better describe domain knowledge. The average depth of main hierarchies was 2,38 with mode of 2 and their skewness showed that there was only a small asymmetry in sample distribution. As expected, the depth of support ontologies was mainly 0, indicating only one supporting class in most hierarchy cases.

Since to all students at least one support hierarchy or class was suggested, it can be concluded that the above result is influenced by this suggestion. With next generation no individual suggestions should be made. Instead, more detailed explanation and more examples of support hierarchies should be included in the teaching process.

Average number of classes was 21,76, but other measures, especially a standard deviation of 17,8309, showed that there are some extreme values (inclining more to greater values, as can be seen from skewness). A number of classes in ontologies that have support classes was also very variable for main classes, but not for support ones. This is understandable, because most of them had only one class, although several of them had as many as 11 or 12.

TABLE II. HIERARCHY ANALYSIS

Criteria	**Raw values**		**Statistical measures**				
	Minimum	***Maximum***	***Average***	***Median***	***Mode***	***Standard deviation***	***Skewness***
Number of hierarchies	1	4	1,62	1	1	0,9452	1,6023
Number of support hierarchies	0	3	0,6	0	0	0,9035	1,5910
Depth of main hierarchies	0	4	2,38	2	2	0,9666	-0,2809
Depth of support hierarchies	0	2	0,31	0	0	0,6806	1,5139
Number of classes	6	110	21,76	16,5	14	17,8309	3,1115
Number of classes in main hierarchies	1	105	22,05	16	16	22,9858	2,6314
Number of classes in support hierarchies	1	12	3,23	2	1	3,4296	1,3704

Results obtained for hierarchy analysis showed that other criteria have to be included for class analysis because of large range of number of classes – from 6 to 110. Since, according to prior notions, students in a certain number of cases tend to represent even instances as classes, this can result in such a large range. Therefore, ontologies with different development mistakes should be analyzed separately. Diversity of the domains represented can be used for grouping of ontologies before ontology analysis.

Hierarchy information could not be affected by number of classes and it showed relatively even distribution. But it also pointed out that beginners do not understand a concept of support classes and their usefulness for better knowledge representation. This ontology feature demands more practice to be exploited.

B. Attributes Analysis

Information about the attributes analysis is presented in Table 3. Criteria used are as follows: total number of attributes, number of attributes in main and support hierarchies (where applicable), number of connecting attributes and number of simple and complex attributes.

Average number of attributes was 12,8, but standard deviation and skewness showed discrepancies of that value. For ontologies with support hierarchies results were the same for attributes used in main hierarchies. In support hierarchies there were no big discrepancies and number of attributes was very small. In most cases there were two attributes (mode value 2), but arithmetic mean of 5,38 and other measures showed variation of attribute number (which was actually from 1 to 17).

A smaller number of attributes in support hierarchies shows that only those for basic description of classes were used (sometimes only instance name). Although those classes help in better description of main hierarchy, the question arises whether they should be also fully described. In that case the description of the main class would also be better. Again, the importance and possibilities that support hierarchies have remain unused.

TABLE III. ATTRIBUTE ANALYSIS

Criteria	**Raw values**		**Statistical measures**				
	Minimum	***Maximum***	***Average***	***Median***	***Mode***	***Standard deviation***	***Skewness***
Number of attributes	4	60	12,18	9	8	9,1377	3,3203
Number of attributes in main hierarchies	2	54	10,13	7,5	8	11,2339	3,4564
Number of attrributes in support hierarchies	1	17	5,38	4	2	3,8580	1,9580
Number of connecting attributes	1	8	3,4	2	2	2,4902	1,0398
Number of simple attributes	2	52	9,4	7,5	8	8,2293	3,4407
Number of complex attributes	0	10	2,78	2	0	2,7575	0,9809

The number of connecting attributes showed that most of ontologies had 2 of them with average of 3,4 and values ranging from 1 to 8. As explained above, connecting attributes can be within main or support hierarchy. More analysis is necessary for determining whether the most often value of 2 attributes in support classes and 2 connecting attributes can indicate the following:

- support hierarchy – one general attribute for defining instance name and the other for reverse connection with the class described with that instance (value of that attribute is the instance of the class which attribute is instance of the class it belongs to – so called reverse slots in Protégé);
- connecting attributes – one attribute in described class and one reverse in class that describes it.

When comparing simple and complex attributes, regardless the values that show asymmetry of the distribution, it is obvious that mostly simple attributes were used. As mentioned above, this was noted during previous years of teaching this course. According to average number of complex attributes, they were probably those used as connection attributes. Obviously, they should be analyzed separately from the rest of complex attributes so that the percentage of usage of each of them can be calculated. Nevertheless, the small number of complex attributes in general showed that all their possibilities for better class description were not used.

In general, high standard deviation and skewness values indicate that distribution asymmetry does not allow accurate results interpretation. Aforementioned problem of representing instances as classes in a certain number of cases can have influence on large number of attributes in some ontologies, underlining that ontologies with different development mistakes should be analyzed separately. After grouping of ontologies according to domain similarity (as suggested in hierarchy analysis) it has to be determined how this will affect attribute analysis results and whether other criteria or ontology manipulation is necessary.

V. Conclusion and Future Work

Results of conducted research pointed out several problems with oversights of new ontology developers. According to evaluated ontologies, common beginner oversights are:

- about 60% of users do not understand the value of support hierarchies in representation of semantic information (with the notion that the result is influenced by individual suggestions to include support hierarchies and that results could have been worse);
- users that created support hierarchies do not exploit their full potential (mostly only one class and less attributes for description of classes in support hierarchies);
- very small number of complex attributes shows that users possibly consider the number of attributes as main feature for embedding semantic information and not their complexity or that they do not fully understand their potential.

Some suggestions for improvement of semantic criteria can also been given, regarding prior analysis:

- the number of classes in general and also in main and support classes – large range in number of classes prevents correct interpretation of results and therefore ontologies with different development mistakes should be analyzed separately with additional semantic criteria;
- grouping of ontologies according to domain similarity can be conducted also with additional semantic criteria;
- the number of attributes – standard deviation shows more or less uneven distribution of values, also disabling correct interpretation, although some general conclusions can be made; after corrections in hierarchy analysis, effects of those changes should be analyzed with possible adjustment of semantic criteria.

Obtained results show that to certain aspects of ontology features more focus should be given when learning or teaching this formalism for representing domain knowledge. The future work in research of this problem will include:

- separation of ontologies with mistakes that cause extreme values in number of classes and/or attributes;
- grouping of ontologies according to domain similarity;
- adjustment of existing and establishment of new criteria;
- trial analysis of 50 ontologies with new settings and full analysis of all ontologies;
- change of focus in ontology development exercises with next generation of students and comparison of results;
- inclusion of second year graduate students that learn Protégé-OWL and description logics with adjustment of semantic criteria.

Given that knowledge representation using ontologies is integral part of Semantic Web and given that incorporating semantics in domain description is a precondition for its success, minimizing oversights that influence on proper representation of semantic information is of high importance. To new ontology developers all features that can aid in this effort should be pointed out.

References

[1] T. Gruber, “A Translation Approach to Portable Ontology Specifications”, Knowledge Acquisition, vol. 5, pp. 199-220, June 1993.

[2] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, Ontological Engineering, London: Springer-Verlag, 2004.

[3] A. Gómez-Pérez, “Evaluation of Ontologies”, International Journal of Intelligent Systems, vol. 16, pp. 391-409, September 2001.

[4] A. Gómez-Pérez, “Ontology Evaluation”, in Handbook on Ontologies, S. Staab and S. Studer, Eds. Berlin: Springer-Verlag, 2004, pp. 251-273.

[5] C. Welty and N. Guarino, “Supporting ontological analysis of taxonomic relationships”, Data & Knowledge Engineering, vol. 39, pp. 51-74, October 2001.

[6] N. Guarino and C. Welty, “Overview of OntoClean”, in Handbook on Ontologies, S. Staab and S. Studer, Eds. Berlin: Springer-Verlag, 2004, pp. 151-171.

[7] A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehmann, “Modelling Ontology Evaluation and Validation”, Proceedings of the 3rd European Semantic Web Conference (ESWC2006), Springer, Budva, Montenegro, June 2006, pp. 140-154.

[8] L. Obrst, W. Ceusters, I. Mani, S. Ray, and B. Smith, “The Evaluation of Ontologies: Toward Improved Semantic Interoperability”, in Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences, C. J. O. Baker and K-H. Cheung, Eds., London: Springer, 2007, pp. 139-158.

[9] Y. Sure et al., “Why Evaluate Ontology Technologies? Because it Works!”, IEEE Intelligent Systems, vol. 19, pp. 74-81, July 2004.

[10] N. F. Noy, R. Guha, and M. A. Musen, “User ratings of ontologies: Who will rate the raters?”, Proceedings of the AAAI 2005 Spring Symposium on Knowledge Collection from Volunteer Contributors, Stanford, CA, March 2005., www.stanford.edu/~natalya/papers/SS505NoyN.pdf, visited July 26, 2010.

[11] S. Lovrenčić and M. Čubrilo, “Ontology Evaluation – Comprising Verification and Validation”, Proceedings of the 19th Central European Conference on Information and Intelligent Systems (CECIIS 2008), FOI, Varaždin, Croatia, Sep. 2008, pp. 657-663.

[12] C. d'Amato, S. Staab, and N. Fanizzi, “On the influence of Description Logics Ontologies on Conceptual Similarity”, in Knowledge Engineering: Practice and Patterns: Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2008), Springer Verlag, Acitrezza, Italy, Sep.– Oct. 2008, pp. 48-63.

[13] S. Lovrenčić, Formal ontology of university studies, Ph.D. Thesis, Varaždin, 2007, in Croatian

[14] ***, Protégé, http://protege.stanford.edu, visited July 26, 2010.

[15] J. H. Gennari et al., “The evolution of Protégé: an environment for knowledge-based systems development”, International Journal of Human-Computer Studies, vol. 58, pp. 89-123, Jan. 2003.

Enriching Ontologies for Named Entity Disambiguation

Hien Thanh Nguyen
Ton Duc Thang University
98 Ngo Tat To St., 19 Ward, Binh Thanh District,
HCM City, Vietnam
hien@tdt.edu.vn

Tru Hoang Cao
HCM City University of Technology
268 Ly Thuong Kiet St., District 10,
HCM City, Vietnam
tru@cse.hcmut.edu.vn

***Abstract*— Detecting *entity mentions* in a text and then mapping them to their right *entities* in a given knowledge source is significant to realization of the semantic web, as well as advanced development of natural language processing applications. The knowledge sources used are often close ontologies - built by small groups of experts - and Wikipedia. To date, state-of-the-art methods proposed for named entity disambiguation mainly use Wikipedia as such a knowledge source. This paper proposes a method that enriches a close ontology by Wikipedia and then disambiguates named entities in a text based on that enriched one. The method disambiguates named entities in a text iteratively and incrementally, including several iterative steps. Those named entities that are identified in each iterative step will be used to disambiguate the remaining ones in the next iterative steps. The experiment results show that enrichment of a close ontology noticeably improves disambiguation performance.**

Keywords- entity disambiguation; ontology enrichment; annotation; named entity; ontology

I. INTRODUCTION

Named entities (NEs) are those that are referred to by names such as people, organizations, or locations. This paper addresses the named entity disambiguation problem (NED) that aims at mapping entity names in a text to right entities in a given source of knowledge. Having been emerging in recent years as a challenging problem, but significant to realization of the Semantic Web, as well as advanced development of Natural Language Processing applications, NED has attracted much attention by researchers all over the world. The problem in reality is that one name in different occurrences may refer to different entities and one entity may have different names that may be written in different ways and with spelling errors. For example, the name "John McCarthy" in different occurrences may refer to different NEs such as a computer scientist from Stanford University, a linguist from University of Massachusetts Amherst, an Australian ambassador, and so on. Such ambiguity makes identifying right entities in a text challenging and raises NED as a key research aspect in the above-mentioned areas.

NED can be considered as an important special case of Word Sense Disambiguation (WSD) [12]. The aim of WSD is to identify which sense of a word is used in a given context when several possible senses of that word exist. In WSD, words to be disambiguated may either appear in a plain text or an existing knowledge base. Techniques for the latter use a dictionary, thesaurus, or an ontology as a sense inventory that defines possible senses of words. Having been emerging recently as the largest and the most widely-used encyclopedia in existence, Wikipedia[1] is used as a knowledge source for not only WSD, but also Information Retrieval, Information Extraction, Ontology Building, Natural Language Processing, and so on [9]. Proposed methods for WSD typically choose a set of features for representation of a target word (or its context) based on features of its surrounding words limited in a window context, and relationships among them and the target word. The context size is commonly set to ±3 or ±5 words around the target word. In recently years, some methods proposed for WSD have been adopted for NED [1][8][13]. When dealing with named entity disambiguation, many works focus on clues in a whole text [3][10][11] for disambiguation, but not just words around the named entity to be disambiguated.

Wikipedia is a free encyclopedia written by a collaborative effort of a large number of volunteer contributors. We describe here some of its resources of information for disambiguation. A basic entry in Wikipedia is a *page* (or *article*) that defines and describes a single entity or concept. It is uniquely identified by its title. In Wikipedia, every entity page is associated with one or more categories, each of which can have subcategories expressing meronymic or hyponymic relations. Each page may have several incoming links (henceforth *inlinks*), outgoing links (henceforth *outlinks*), and *redirect* pages. A redirect page typically contains only a reference to an entity or a concept page. Title of the redirect page is an alternative name of that entity or concept. For example, from redirect pages of the United States, we extract alternative names of the United States such as "US", "USA", "United States of America", etc. Other resources are disambiguation pages. They are created for ambiguous names, each of which denotes two or more entities in Wikipedia. Based on disambiguation pages one can detect all entities that have the same name in Wikipedia.

In literature, the knowledge sources used for NED can be divided into two kinds: close ontologies and open ontologies. Close ontologies are built by experts following a top-down approach, with a hierarchy of concepts based on a controlled vocabulary and strict constraints, e.g., KIM [17], WordNet [18]. These knowledge sources are generally of high reliability, but their size and coverage are restricted. Furthermore, not only is the building of the sources labor-intensive and costly, but also they are not kept updated of new discoveries and topics that arise daily. Meanwhile, open ontologies are built by collaborations of volunteers following a bottom-up

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

approach, with concepts formed by a free vocabulary and community agreements, e.g. Wikipedia. Many open ontologies are fast growth with wide coverage of diverse topics and keeping up date daily by volunteers, but someone has doubt about quality of their information contents. Wikipedia is considered as an open ontology where contents of its articles have high quality. Indeed, in [21], Giles investigated the accuracy of content of articles in Wikipedia in comparison to those of articles in Encyclopedia Britannica, and showed that both sources were equally prone to significant errors.

While state-of-the-art NED methods mainly use Wikipedia as the target knowledge source, there are still many application systems based on close ontologies. This paper thus focuses on mapping entity mentions in a text to a close ontology. It faces the following difficulties:

- Those methods proposed for NED using Wikipedia are not easy to adopt to close ontologies because they exploit Wikipedia-based features which do not appear in the close ontologies.
- While information describing entities in Wikipedia is diverse and rich, information describing entities in a close ontology is poor and mainly based on a given number of built-in properties of the entities in that ontology.

Therefore, for automatic mapping entity mentions in a text to a close ontology (henceforth *ontology*), we do need a new method to overcome the above-mentioned difficulties. This paper proposes a method that disambiguates named entities in a text using an ontology where descriptions of entities in that ontology are enriched by features extracted from Wikipedia. The contributions of our proposed method are as follows. First, the method enriches information describing entities in an ontology by their features extracted from Wikipedia, and then disambiguates named entities in a text based on that enriched ontology. Second, the method disambiguates named entities in a text iteratively and incrementally, including several iterative steps. Those named entities that are identified in each iterative step will be used to disambiguate the remaining ones in the next iterative steps. Third, the experiment results show that features extracted from Wikipedia to enrich representation of entities in an ontology noticeably improve disambiguation performance in comparable with not using those features.

The rest of this paper is organized as follows. Section 2 presents our statistical ranking model. Section 3 presents a process of ontology enrichment. Section 4 presents the proposed method for NED. Section 5 presents experiment results. Section 6 presents related works and a conclusion is drawn in Section 7.

II. A Proposed Statistical Ranking Model

In this section, we present a statistical ranking model where we employ the Vector Space Model (VSM) to represent *ambiguous*[2] mentions and entities in a given knowledge source by their features. The VSM considers the set of features of each entity or mention as a ‘bag of words’. We present how each bag of words is normalized. Then we present how to weight words in the VSM and calculate the similarity between feature vectors of mentions and entities. Based on the calculated similarity, our disambiguation method ranks the candidate entities of each mention and chooses the best one. The quality of ranking depends on used features.

Normalization

After extracting features for a mention or an entity, we put them into a ‘bag of words’. Then we normalize the bag of words as follows: (i) removing special characters in some tokens such as normalizing U.S to US, D.C (in “Washington, D.C” for instance) to DC, and so on; (ii) removing punctuation mark and special tokens such as commas, periods, question mark, $, @, etc.; (iii) removing stop words such as *a*, *an*, *the*, etc.; and (iv) stemming words using Porter stemming algorithm.

After normalizing the bag of words, we are already to convert it in to a token-based feature vector.

Term weighting

For a mention, suppose there are *N* candidate entities for it in a given knowledge source. We use the *tf-idf* weighting schema viewing each ‘bag of words’ as a document and using cosine similarity to calculate the similarity between the bag of words of the mention and the bag of words of each of the candidate entities respectively. Given two vector S_1 and S_2 for two bags of words, the similarity of the two bags of words is computed as:

$$Sim(S_1, S_2) = \sum_{common\ word\ t_j} w_{1j} * w_{2j}$$

where t_j is a term present in both S_1 and S_2, w_{1j} is the weight of the term t_j in S_1 and w_{2j} is the weight of the term t_j in S_2.

The weight of a term t_j in vector S_i is given by:

$$w_{ij} = log(tf_j{+}1).log(N/df_j)/\sqrt{s_{i1}^2 + s_{i2}^2 + ... + s_{iN}^2} \qquad (1)$$

where tf_j is the frequency of the term t_j in vector S_i, N is the total number of candidate entities, df_j is the number of bags of words representing candidate entities in which the term t_j occurs, $s_{ij} = log(tf_j{+}1)\ .log(N/df_j)$.

Algorithm

For a mention *m* that we want to disambiguate, let *C* be the set of its candidate entities. We cast the named entity disambiguation problem as a ranking problem with the assumption that there is an appropriate scoring function to calculate semantic similarity between feature vectors of an entity $c \in C$ and the mention *m*. We build a ranking function that takes as input the feature vectors of the entities in *C* and the feature vector of the mention *m*, then based on the scoring function to return the entity $c \in C$ with the highest score. We use *Sim* function as given in Equation 1 as the scoring function. What we have just described is implemented in Algorithm 1. *Sim* is used at Line 3 of the algorithm.

[2] An *ambiguous* mention is a mention that is used to refer to two or more entities in a given knowledge source. We call these entities *candidate entities* of that mention.

Algorithm 1 Statistics-based Entity Ranking

```
let $C$ a set of candidate entities of $m$
for each candidate $c$ do
    $score[c] \leftarrow Sim(FeatureVector(c), FeatureVector(m))$
end for
$c^* \leftarrow \arg\max_{c_i \in C} score[c_i]$
if $score[c^*] > \tau$ then return $c^*$
return $NIL$
```

III. Ontology Enrichment

Usually, a built-in ontology in a system does not represent enough information about NEs, which causes misclassification and mis-identification of NEs referred to in a text with respect to that ontology. There are two kind of missing information of entities in an ontology. First, the ontology defines not enough properties of many entities. For instance, persons in PROTON ontology are represented by only four properties *hasPosition*, *hasProfession*, *hasRelative* and *isBossOf*. In reality, a person has a lot of different relations with other entities such as relation to persons other than relatives (e.g., Hillary Clinton, wife of Bill Clinton), or notable achievements (e.g., John McCarthy, inventor of LISP), etc. Second, some properties of a certain entity may be not assigned values.

To overcome these shortages of a close ontology, we need to enhance representations of entities in that ontology to enrich their attributes and relations by new features from another source of knowledge. In particular, in this paper, we exploit Wikipedia to generate features whose values provide additional information about focused NEs, such as location where one was born, or fellow-workers, etc., for enriching representation of NEs in a given ontology by an enrichment process. Then the disambiguation is performed using that enriched ontology. Such enrichment leads to representations of those entities in a richer space, which facilitates employment of a statistical model for disambiguation.

Before performing enrichment, entities in Wikipedia and in the ontology are already represented by their features. We call features extracted from the ontology for representing entities in it *ontology features* (OF). We call features extracted from Wikipedia for representing Wikipedia entities *Wikipedia features* (WF). Here we describe the features.

Ontology features

We utilize ontological concepts, and properties of entities in a specific ontology to extract their features. In particular, let I be a set of entities of an ontology $\mathcal{O}$; for each entity $i \in I$, the following features are extracted to represent it: (1) all classes to which i belongs; (2) attribute values of i; and (3) all names and identifiers of entities that have relationship with i or vice versa.

Wikipedia features

For each entity in Wikipedia, serving as a candidate entity for an ambiguous mention in a text, we extract the following information to construct its feature vector.

– *Entity title* (ET). Each entity in Wikipedia has a title. For instance, "John McCarthy (computer scientist)" is the title of the page that describes Professor John McCarthy who is the inventor of LISP programming language. We extract "John McCarthy (computer scientist)" for the entity Professor John McCarthy.

– *Titles of redirect pages* (RT). Each entity in Wikipedia may have some redirect pages whose titles contain different names, i.e., aliases, of that entity. To illustrate, from the redirect pages of an entity John Williams in Wikipedia, we extract their titles: Williams, John Towner; John Towner Williams; Johnny Williams; Williams, John; John Williams (composer); etc.

– *Category labels* (CAT). Each entity in Wikipedia belongs to one or more categories. We extract labels of all its categories. For instance, from the categories of the entity `John McCarthy (computer scientist)` in Wikipedia, we extract the following category labels as follows: Turing Award laureates; Computer pioneers; Stanford University faculty; Lisp programming language; Artificial intelligence researchers; etc.

– *Outlink labels* (OL). In the page describing an entity in Wikipedia there are some links pointing to other Wikipedia entities. We extract labels (anchor texts) of those outlinks as features of that entity.

– *Inlink labels* (IL). For an entity in Wikipedia, there are some links from other Wikipedia entities pointing into it. We extract labels of those inlinks as its possible features.

After extracting features for entities in Wikipedia and a given ontology, we put them into 'bag of words'. Then the bag of words are normalized and converted to feature vectors. Now we are ready to present the enrichment algorithm.

Enrichment Algorithm

We present steps that enrich representation of an entity $i \in I$ in an ontology $\mathcal{O}$ as follows:

– Step 1: The longest name of i, namely n, is used as a query to retrieve candidate entities from Wikipedia.

– Step 2: If the number of candidate entities in the returned set is higher than 1, go to Step 5; otherwise, go to Step 3.

– Step 3: If the number of candidate entities in the returned set is 1, that only one entity, namely c, is checked to be sure that it is the same as i. In particular, let R_i be a set of entities that have relationship with i in the ontology and W_c be a set of entities that have relationship with c in Wikipedia; if R_i is a subset of W_c, then i and c are considered as the same referent.

– Step 4: If there are not any entity in the returned set, prefixes and postfixes (e.g., Mr., company, inc., co., etc.) of n are removed. Then n becomes n'. Go to Step 2. For instance, if using "Columbia Sportswear Company" to retrieve candidate entities and the returned set is empty, the postfix "Company" is removed and then "Columbia Sportswear" is used as a query.

– Step 5: When the number of candidate entities in the returned set is higher than 1, Algorithm 1 is applied to rank the candidate entities. The candidate entity with the highest rank is chosen and its features are used to enrich representation of the corresponding entity in $\mathcal{O}$**.** Note that this algorithm does not exploit identifiers of entities in $\mathcal{O}$ as their features**.**

These steps are applied to enrich all entities in $\mathcal{O}$. Then we obtain a new ontology whose entity representations are enriched. Note that the feature generation and enrichment is performed prior to NE disambiguation, and is completely independent of the later steps; therefore, it can be built once and reused for NE disambiguation tasks in the future.

IV. Named entity disambiguation

We recall that the method this paper proposes to NED is to map entity mentions in a text to right entities a close ontology $\mathcal{O}$. After ontology $\mathcal{O}$ is enriched by Wikipedia, we obtain an enriched ontology $\mathcal{O}_e$**.** Then the method performs disambiguation based on $\mathcal{O}_e$. Each entity in $\mathcal{O}_e$ is represented by the features OF and WF as described above. To map a mention in a text to the right entity in $\mathcal{O}_e$, our method extracts features in the text to represent that mention. We call these features *text features* and describe them below.

Text features

To construct the feature vector of a mention in a text, we extract all mentions co-occurring with it in the whole text, local words in a context window, and words in the context windows of those mentions that are co-referent with the mention to be disambiguated. Those features are presented below.

- *Entity mentions* (EM). After named entity recognition, mentions referring to named entities are detected. We extract these mentions in the whole text.
- *Local words* (LW). All the words found inside a specified context window around the mention to be disambiguated. The window size is set to 55 words, not including special tokens such as $, #, ?, etc., which is the value that was observed to give optimum performance in the related task of cross-document coreference resolution ([6]). Then we remove those local words that are part of mentions occurring in the window context to avoid extracting duplicate features.
- *Coreferential words* (CW). All the words found inside the context windows around those mentions that are co-referent with the mention to be disambiguated in the text. For instance, if "John McCarthy" and "McCarthy" co-occur in the same text and are co-referent, we extract words not only around "John McCarthy" but also those around "McCarthy". The size of those context windows are also set to 55 words. Note that, when the context windows of mentions that are co-referent are overlapped, the words in the overlapped areas are extracted only once.
- *Identifiers* (ID). All identifiers of identified entities in a text are features.

Disambiguation

The proposed method in this paper disambiguates named entities in text iteratively and incrementally, including several iterative steps. Those named entities that are identified in each iterative step will be used to disambiguate the remaining ones in the next iterative steps. In other words, we exploit identifiers of identified entities in the text as extended parts of that text. These identifiers are used as features of the remaining ones.

Algorithm 2 implements the method. The loop statement at Line 3 stops when the set of identified entities *E* has no change between two iteration steps or all mentions are mapped to an entities ontology $\mathcal{O}_e$. Line 7 call Algorithm 1 to rank candidate entities of a mention. The *revised* function at Line 9 adjusts *E* using the coreference chain of a mention. For example, assume that in a text there are occurrences of coreferent mentions "Denny Hillis" and "Hillis; if "Denny Hillis" is recognized as referring to W. Daniel Hillis in Wikipedia for instance, then "Hillis" also refers to W. Daniel Hillis.

Algorithm 2 Iterative and Incremental Disambiguation

```
1:  let $\mathcal{N}$ be a set of mentions and $E$ be an *empty* set
2:  *flag* ← **false**
3:  **loop until** $\mathcal{N}$ *empty* or *flag is* ***true***
4:      $\mathcal{N}' \leftarrow \mathcal{N}$
5:      **for each** $n \in \mathcal{N}'$ **do**
6:          $C \leftarrow$ a set of candidate entities of $n$
7:          $\gamma^* \leftarrow$ run Algorithm 1 for $n$
8:          **If** $\gamma^*$ *is not NIL* **then**
9:              map $n$ to $\gamma^*$
10:             $E \leftarrow revised(E \cup \{n \rightarrow \gamma^*\})$
11:             remove $n$ from $\mathcal{N}$
12:         **end if**
13:     **end for**
14:     **if** *E no change* **then** *flag* = **true**
15: **end loop**
```

We note that a coreference chain might not be correctly constructed in the pre-processing steps due to the employed NE coreference resolution module. Moreover, for a correct coreference chain, if there is more than one mention already resolved, then it does matter to choose the right one to be propagated. Therefore, for a high reliability, before propagating the referent of a mention that has already been resolved to other mentions in its coreference chain, our method checks whether that mention satisfies one of the following criteria: (i) The mention occurs in the text prior to all the others and is one of the longest mentions in its coreference chain, or (ii) The mention occurs in the text prior to all the others in its coreference chain and is the main alias of the corresponding entity in the ontology. Regarding the computational cost, since after each iteration of the outer loop there is at least one more mention resolved or *E* has no change, the worst case complexity is $O(N^2)$, where *N* is the number of mentions to be resolved.

V. EVALUATION

First of all, we perform enrichment of KIM ontology by Wikipedia using the ontology enrichment algorithm presented in Section 3. For experiments, we build a dataset by collecting documents that contain mentions of entities in KIM ontology. All mentions are manually mapped to that ontology to form a golden standard corpus.

There are total 186 documents in the dataset. Table 1 presents information about the mentions that contain “Georgia” or “Columbia” in the dataset. The right column in the table shows the number of those mentions in the dataset referring to the corresponding entity in the left column. For instance, as showed in the second row of the table, there are 90 mentions referring to the entity Georgia – a state of the United States.

Since we aim at evaluating how good our method is in terms of disambiguation performance, we focus on ambiguous mentions. Therefore, in order to produce ambiguous mentions for the experiments, we replace each mention containing “Georgia” by only “Georgia” and each mention containing “Columbia” by only “Columbia”. For instances, we replace “South Georgia and the South Sandwich Islands” by “Georgia”, “Columbia University” by “Columbia”, etc.

TABLE I. STATISTICS ABOUT AMBIGUOUS MENTIONS IN THE DATASET

Entity	# of mentions
Georgia (country)	318
Georgia (U.S. state)	90
South Georgia and the South	59
British Columbia	34
Columbia Sportswear Company	65
Columbia University	13
Columbia, South Carolina	15
Space Shuttle Columbia	80
District of Columbia	1
Total	675

TABLE II. STATISTICS ABOUT TOTAL AMBIGUOUS MENTIONS AND DISAMBIGUATED MENTIONS

Mention	# of candidate entities	# of total mentions	# disambiguated mentions
Georgia	7	468	463
Columbia	10	207	205
Total		675	668

Note that prior to disambiguation, we perform preprocessing tasks. In particular, we perform NE recognition and NE coreference resolution using natural language processing resources of Information Extraction engine based on GATE [5]. The NE recognition applies pattern-matching rules written in JAPE’s grammar of GATE to detect and tag boundaries of mentions occurring in the dataset and then categorize corresponding entities as Person, Location and Organization, etc. After detecting all mentions occurring in the text, we run NE co-reference resolution [2] module in the GATE system to resolve the different mentions of a NE into one group that uniquely represents the NE. After that we run Algorithm 2 for disambiguation. In [16], the authors explored a range of features extracted from texts and Wikipedia, and vary combinations of those features to appraise which ones are good for NED. It shows that the Wikipedia features ET, RT, CAT and OL in combination with the text features EM, LW and CW give the best performance. Based on that finding, when conducting experiments, we focus on the combination OF + ET + RT + CL + OL with regard to Wikipedia features. Table 2 shows the number of candidate entities, the number of total ambiguous mentions and the number of disambiguated mentions.

We test the method in two settings of entity representation using the basic features extracted from the given ontology (i.e., OF) and using those basic features in combination with features extracted from Wikipedia (i.e., OF + ET + RT + CL + OL on the enriched ontology). Table 3 shows the experiment results in these settings. The third column of Table 3 shows the number of correct mappings of mentions in the dataset to their corresponding entities in the ontology. The results show that the features extracted from Wikipedia in combination with the basic features noticeably improve disambiguation performance in comparison with using the basic features only.

TABLE III. DISAMBIGUATION PERFORMANCE IN TERMS OF PRECISION AND RECALL

Mention	Features	# of correct mappings	P (%)	R (%)
Georgia	OF	310	66.95	66.23
	OF + ET + RT + CL + OL	436	94.16	93.16
Columbia	OF	171	83.41	82.60
	OF + ET + RT + CL + OL	183	89.26	88.40
Total	OF	481	72.00	71.25
	OF + ET + RT + CL + OL	619	92.66	91.70

VI. RELATED WORKS

There are many methods proposed for NED in literature. Methods disambiguating named entities based on Wikipedia are overwhelming. The method in [19] relies on affiliation, text proximity, areas of interest, and co-author relationship as clues for disambiguating person names in calls for papers only. Meanwhile, the domain of [20] is that of geographical names in texts. The authors use some patterns to narrow down the candidates of ambiguous geographical names. For instance, “*Paris, France*” more likely refers to the capital of France than a small town in Texas. Then, it ranks the remaining candidate entities based on the weights that are attached to classes of the constructed Geoname ontology. The method in [13] generates a co-occurrence model from article’s templates that served as training data and then employed the SVM for place-name disambiguation. This method only works on co-occurrence place-names. It chooses a window size of ±10 location references regardless of other words that are not part of place-names. In contrast, the problem that we address in this paper is more general, which is not limited to named entities of a particular class or domain, but for all that may occur in a text.

In [8], authors implemented and evaluated two different disambiguation algorithms that extracted terms in a document and linked them to Wikipedia articles using Wikipedia as a sense inventory. Then they reported the best performing algorithm was the one using a supervised learning model where Wikipedia articles, which had already been annotated, served as training data. This algorithm used the local context of three words to the left and right, with their parts-of-speech, as features for representing an ambiguous term. In 2007, we proposed an idea of exploiting identified entities to disambiguate remaining ones [14]. Later on, in 2008, the works in [10] bore a resemblance to our idea for disambiguating terms in a documents using Wikipedia. The works in [11] extended both works [8] and [10] by exploiting relatedness of a target term to its surrounding context, besides exploiting the feature as in the latter one.

The works in [1] and [3] exploit several of the disambiguation resources such as Wikipedia articles (entity pages), redirection pages, categories, and links in the articles. The methods in [1] extracted words inside a 55-word window around a mention to form its feature vector. Based on the cosine similarity between feature vectors, they ranked candidate entities for a mapping and chose the one with the highest similarity score. Due to too low similarity scores with the cosine-based ranking in many cases, the authors employed the Support Vector Machine model (SVM) to learn a mapping from the context window to the specific categories of articles. The method in [3] exploited the same resources of information in Wikipedia for the disambiguation task as in [1]. This method simultaneously disambiguates all mentions in a document by maximizing the agreement among categories of candidate entities and maximizing the contextual similarity between contextual information in the document and context data stored for the candidate entities. The context data comprise appositives in the titles of articles and phrases that appear as anchor texts of links in the first paragraphs of the articles. The contextual information of a document contains all phrases occurring in the context data. The method in [15] exploited ET, CAT, OL and the most frequency words in each Wikipedia article to represent entities in Wikipedia. Then it calculated semantic relatedness using a random walk model for simultaneously disambiguating all mentions in a document.

VII. Conclusion

We proposed a method that enriches a close ontology and then disambiguates named entities in a text based on that enriched one. Our proposed disambiguating method is iteratively and incrementally, including several iterative steps. Those named entities that are identified in each iterative step will be used to disambiguate the remaining ones in the next iterative steps. The experiment results show that disambiguating named entities based on an ontology enriched by Wikipedia noticeably improves disambiguation performance in comparison with that of disambiguation based on the original ontology. Our method solves the problems of named entity disambiguation on a close ontology with poor entity descriptions and limited number of entity properties.

References

[1] R. Bunescu and M. Paşca, “Using encyclopedic knowledge for named entity disambiguation,” in Proc. of the 11th Conference of EACL, 2006, pp. 9–16.

[2] K. Bontcheva, M. Dimitrov, D. Maynard, V. Tablan, and H. Cunningham, “Shallow methods for named entity coreference resolution,” in Proc. of TALN 2002 Workshop, 2002.

[3] S. Cucerzan, “Large-scale named entity disambiguation based on Wikipedia data,” in Proc. of EMNLP-CoNLL 2007, 2007, pp. 708–716.

[4] W. Cohen, P. Ravikumar, and S. Fienberg, “A comparison of string distance metrics for name-matching tasks,” in IJCAI-03 II-Web Workshop, 2003, pp. 73-78.

[5] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, “GATE: A framework and graphical development environment for robust NLP tools and applications,” in Proc. of ACL’02, 2002, pp.168-175.

[6] C. H. Gooi and J. Allan, “Cross-document coreference on a large-scale corpus,” in Proc. of HLT/NAACL’04, 2004, pp. 9-16.

[7] R. Mihalcea, “Using Wikipedia for automatic word sense disambiguation,” in Proc. of HLT/NAACL’07, 2007, pp. 196–203.

[8] R. Mihalcea and A. Csomai, “Wikify!: Linking documents to encyclopedic knowledge,” in Proc. of CIKM’07, 2007, pp. 233–242.

[9] O. Medelyan, D. Milne, C. Legg, and I. H. Witten, “Mining meaning from Wikipedia,” International Journal of Human-Computer Studies, 67(9), 2009, pp. 716-754.

[10] O. Medelyan, I. H. Witten and D. Milne, “Topic indexing with Wikipedia,” in Proc. of WIKIAI’08.

[11] D. Milne and I. H. Witten, “Learning to link with Wikipedia,” in Proc. of CIKM’08, 2008, pp. 509–518.

[12] R. Navigli, “Word sense disambiguation: A Survey,” ACM Computing Surveys, 41(2), 2009, pp. 1-69.

[13] S. Overell and S. Rüger, “Using co-occurrence models for placename disambiguation,” The IJGIS, Taylor and Francis, 2008, pp. 265-287.

[14] H. T. Nguyen and T. H. Cao, “A Knowledge-based approach to named entity disambiguation in news articles,” in Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), Springer, vol. 4830, 2007, pp. 619–624.

[15] A. Gentile, Z. Zhang, L. Xia and J. Iria, “Semantic relatedness approach for named entity disambiguation,” in Proc. of 6th Italian Research Conference on Digital Libraries - IRCDL 2010, 2010.

[16] H. T. Nguyen and T. H. Cao, “Exploring Wikipedia and text features for named entity disambiguation,” in: N.T. Nguyen, M.T. Le, and J. Świątek (Eds.): ACIIDS 2010, Part II, LNCS, Springer, vol. 5991, 2010, pp.11–20.

[17] A. Kiryakov, B. Popov, I Terziev, D. Manov, and D. Ognyanoff, “Semantic annotation, indexing, and retrieval,” Journal of Web Semantics, 2(1), 2005, pp.49-79 .

[18] G. A. Miller, “WordNet: A lexical database for English,” Communications of the ACM, 38, 1995, pp.39–41.

[19] J. Hassell, B. Aleman-Meza, and I. B. Arpinar, “Ontology-Driven Automatic Entity Disambiguation in Unstructured Text,” in Proc. of ISWC2006, 2006, pp. 44–57.

[20] V. Raphael, K. Joachim, and M. Wolfgang, “Towards Ontology-based Disambiguation of Geographical Identifiers,” in Proc. of the 16th WWW Workshop on I3: Identity, Identifiers, Identifications, 2007.

[21] Jim Giles, “Internet encyclopaedias go head to head,” Nature 438 (7070), 2005, pp. 900-901.

Temporal Aspects in Diagnosis Validation

Manuela Popescu*, Pascal Lorenz**, Marc Gilg**, Jean Marc Nicod*
*) University of Besançon, France
**) University of Haute Alsace, France
manuela.popescu@univ-fcomte.fr, lorenz@ieee.org, marc.gilg@uha.fr, jean-marc.nicod@lifc.univ-fcomte.fr

Abstract— **In this article, we introduce temporal aspects related to diagnosis validation. Event correlation and action triggering are essential for an accurate diagnosis decision. There are several time-related challenges referring to event timestamps, timely event correlations, and timely corrective actions, in both absolute time (precise moment), or relative time (between events, actions, and events and actions). We propose here a new timestamp approach and we consider a series of temporal operators defining the event relative temporal position that allows a more fine grain interpretation of the system behavior. A combination of proposed mechanisms is used to complete the main functions of a diagnosis engine.**

Keywords- ***diagnosis validation; timestamps ; temporal features; temporal actions; temporal logics.***

I. INTRODUCTION

The complexity of networks and distributed systems gives rise to management challenges when unexpected situations occur. There is an overwhelming number of feedback events coming from the system in the form of status reports towards the monitoring and management applications and human operators. Actually, very few of these events, less than 10%, can be considered for potential status understanding and remedy. Given the numbers, it is inevitable that many relevant events are dropped. The remedy actions can come too late (and sometimes be useless). There are numerous management applications in commercial use. However, the variety of the systems to be managed, their complexity, and the fact that most of the successful decisions are rarely recorded, rise serious challenges in the ability to accurately handle unexpected situations.

Some of the multiple causes leading to the current state are (i) lack of successful validation of corrective actions, (ii) heterogeneity of the events to be handled, and (iii) incomplete correlation and time synchronization between status reports, decision processing and corrective actions.

To address the lack of successful validation of corrective actions, two loops of the diagnosis process were identified in [1]: (a) one loop deals with measuring the system parameters (system state, events, *i.e.,* pre-conditions) and takes the most suitable actions; this was referred to as the *diagnosis loop* (b) a second loop deals with validating that the corrective actions were indeed successful; this was referred to as the *validation loop*. The main goals of the validation loop were (a) to establish the new state of the system, *i.e.*, post-conditions and (b) to gather knowledge on how to solve future similar situations, in case the actions taken were considered successful. In addition, through the concept of *Quality of Diagnosis (QoD)* introduced into the validation loop, the accuracy of the corrective actions and their use in similar situations were enhanced.

A step towards automated diagnosis was introduced in [2], where an event ontology and a progressive diagnosis ontology were proposed. Event dependencies captured by ontology and specific event relations have been formalized. Probable cause and recommended actions were associated with events. Additionally, an augmented specification for actions was proposed to help the validation loop. Both proposals had as a target the reuse of knowledge for problem fixing, identification of recommended diagnosis actions, and validation of successful actions.

The third identified challenge is time-related; this refers to event timestamps, timely event correlations, and timely corrective actions, in both absolute time (precise moment), or relative time (between events, actions, and events and actions). This aspect is more difficult, as many events issued at different timestamps might be processed for event compression/aggregation. The correct adoption of temporal aspects can solve potential conflicts among the post-conditions of the actions already validated as "successful" and helps evaluate the accuracy of the diagnosis actions (preciseness versus permanent damage).

In this paper, we highlight the relevance of temporal aspects, identify the challenging issues, and propose a new timestamp approach. We consider a series of temporal operators defining event relative temporal position that allows a more fine grain interpretation of the system behavior. A combination of proposed mechanisms is used to complete the main functions of a diagnosis engine.

The article has the following structure: Section II presents the state of the art with respect to temporal considerations. In Section III, we talk about approaching temporal aspects. Section IV describes the use of temporal aspects for diagnosis. Section V presents the conclusion and future work.

ISBN: 978-1-61208-104-5

II. STATE OF THE ART

Temporal features are related to several generic aspects concerning (i) inaccurate (wrong, un-synchronized, or missing) clocks, (ii) loss of events, and (iii) hierarchical event processing at layers exposing different clocks. These are somehow related to event propagation skew but also to different syntactic and semantic implementation decisions of the timestamps (including time zones). One approach in dealing with real-time measurements of propagation skew uses a statistical evaluation to update the timer values [6].

Some diagnostic constraints might be temporal. In [2], temporal constraints are used for event tags to define the event ontology and to detect the relative temporal constraints. Walzer *et al.* use specific operators for time-intervals with quantitative constraints in rule-based systems to trigger certain actions [7]. In the following sections, we present the main approaches used to specify temporal aspects on events and actions.

A. Temporal aspects for events

Timestamps are usually carried by the events themselves; basic events possess special timestamp fields that are instantiated when an event instance occurs. Timestamps are storing time in the native format of the platform in which the event processing runs. There are two standard ways to represent the time: (i) using the universal time, or (ii) using time zones. Since one still needs to preserve the zone indication for a device for hourly performance reports, the representation in the universal time is only for the computational point of view. Another standard way to represent the time is the UNIX-format time as a four-byte integer that represents the seconds elapsed since January 1, 1970. For the same reasons, the time zone of the source device should be stored.

An event might have multiple timestamps; the source timestamp (not always present), the logging host timestamp, the console timestamp, and the processing timestamp. Temporal correlation and event aggregation should consider all these timestamps.

Event processing and correlation need a time-based logic to express the relative position of start / end /duration of the events [3]. While attempts were identified for classifying the relative position of the events, no particular commercial solutions are known where a full range of temporal situations are used.

B. Temporal aspects for actions

An enhanced action model was proposed in [2]. One temporal aspect is related to the triggering condition (guard). Others temporal aspects are related to the temporal dependencies between actions, *i.e.*, some action must start at a given period after one action was triggered or was deemed successfully finished.

A diagnosis-oriented augmented action definition was introduced in [2], as follows.

action::= <<guard><ID><post-conditions>
<mode><conflicting>,
where

ID::= READ | WRITE | DELETE | CHANGE | , etc.

mode ::= <potential | recommended | successful <context>>,
with
potential: any diagnosis action that is designated as being related to a potential domain
recommended: any potential action that is perceived as solving a given problem, eventually based on a diagnoses history
successful: when post-conditions were validated as true
context: <d:D, c:C>
d:D is d instance of Domain
c:C is c instance of Cloud

Also in [2], we associated the notion of “conflicting” with a given *action*, which designates the actions a potential action is in conflict with, in a given domain:

conflicting ::= <a_1, a_2,… a_k | a_i:*A*>

A <guard> is acting as pre-conditions and igniter (initial timestamp), and the <post-conditions> are expected to be true (after the action is considered successfully performed). In general, actions are applied following a simple rule:

IF <pre-conditions>
THEN <action> WITH <post-conditions>

Post-conditions are assumed to hold. A composition of actions, a plan, is a set of related actions and it is used to specify dependencies between actions. This is schematically represented in Figure 1. The model can be summarized as follows, where a plan is introduced as a temporal combination of atomic actions (see ID above) [8].

policy::= IF <pre-cond> THEN {<> 1<action> 1<plan>}
[ELSE {<> 1<action> 1<plan>} <action> 1<plan>}]
«post-cond>]

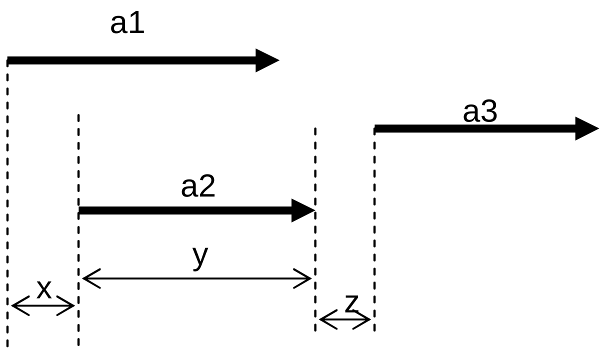


Figure 1. A plan ─ actions: a_1, a_2, a_3; time durations: x, y, z

Based on the analysis of the state of the art, we conclude that there is a need for a unified timestamps approach and a set of operators that must be used in synchronism to express the dependency between events, between actions, or between events and actions [4][5].

In the following sections, we propose a representation of temporal features allowing various semantics used to correlate the events and the actions.

III. APPROACHING TEMPORAL ASPECTS

This section describes aspects related to timestamps, event correlation with temporal operators and gives an example of use of temporal operators.

A. Timestamps

In a hierarchical model, an event model should allow multiple timestamps, depending on the event hosting and processing. In an XML-like specification, we introduce for the device (source), host (server), and processing application (management application or console), the timestamp and the time zone a source, host or processing application belongs to.

TABLE I: Timestamp specification

```
<time>
<device_time> device_time></device_time>
<device_zone> device_time_zone</device_zone>
<server_time> server_time</server_time>
<server_zone> server_time_zone</server_zone>
< processor_time> event_processor_time</
processor_time>
<processor_zone> processor_time_zone</processor_zone>
</time>
```

The timestamp of the event is best set by the event producer (*device_time*). The timestamp representing the moment of event registration on the server, *server_time* is of relevance for correlation. Finally, the timestamp of the entity performing correlation or event processing is relevant for synchronization among multiple such event processing systems.

Any of these three entities can belong to different time zones that should be considered when temporal priorities count.

The values of these parameters are set by various entities. Some protocols provide the capability to supply the time in the occurred event, or the time when the event producer sent the event. With the Network Time Protocol (NTP) the time from event producers will be the most accurate. Alternatively, the time registered by the event processing system might be considered.

We advocate the following representation, similar to Syslog protocol, *e.g.*, *device_time: Jan 1 14:22:45* represents the local time on the device at the time the message is signed. For devices with no clocks, *device_time: Jan 1 00:00:00* should be the representation.

B. Event correlation with temporal operators

Temporal relations are used to build time-dependent event correlations between events. For instance, we may correlate the alarms that happened within the same 10-minutes period, which means the correlation window is 10 minutes. We abstract an event and consider only the temporal aspects.

Let ***e1*** and ***e2*** be two events defined on a time interval:

$T_1 = [t_1, t_1']$
$T_2 = [t_2, t_2']$
and e_1 within T_1
e_2 within T_2

two events occurring within the time intervals T_1 and T_2, respectively.

The following temporal relations R(*t*) or R are identified:

R(*t*):: = {**after**(*t*), **follows**(*t*), **before**(*t*), **precedes(***t***)**}

R ::= {**during**, **starts**, **finishes**, **coincides**, **overlaps**}

The following deductions hold:

after: e_2 ***after(t)*** $e_1 \Leftrightarrow t_2 > t_1+t$

follows: e_2 ***follows(t)*** $e_1 \Leftrightarrow t_2 \geq t_1'+t$

before: e_2 ***before(t)*** $e_1 \Leftrightarrow t_1' \geq t_2'+t$

precedes: e_2 **precedes(t)** $e_1 \Leftrightarrow t_1 \geq t_2'+t$

during: e_2 ***during*** $e_1 \Leftrightarrow t_2 \geq t_1$ *and* $t_1' \geq t_2''$

starts: e_1 ***starts*** $e_2 \Leftrightarrow t_1 = t_2$

finishes: e_1 ***finishes*** $e_2 \Leftrightarrow t_1' = t_2'$

coincides : e_2 ***coincides*** *with* $e_1 \Leftrightarrow t_2 = t_1$ *and* $t_1' = t_2'$

overlaps: e_1 ***overlaps(ε)*** $e_2 \Leftrightarrow t_2' \geq t_1' \pm \varepsilon > t_2 \geq t_1 \pm \varepsilon$
where ε is the accepted threshold for
measurement variation.

With respect to the algebraic properties of the temporal relations,

- all are transitive, except **overlaps**,
- **starts**, **finishes**, **conincides** are also symmetric relations.

C. Example of using temporal operators

In [1], time-oriented diagnosis was defined as

$$[e_1, e_2, e_3 \ldots e_n]t_1 \rightarrow \{p_i\}t_1 \rightarrow \{d_i\}t_1,$$

where

p_i, d_i, and e_i represent a given instance of a problem, diagnosis, and event, respectively.

As an example, let us consider the instantiation:

$\{[e_1, e_2, e_3] \mid e_2$ ***follows***(x) e_1 & e_2 ***overlaps(ε)*** $e_3\}$
$\rightarrow p_{123} \rightarrow d_{123}$
where x is the time duration between e_1 and e_2.

As a note,
$\{[e_1, e_2, e_3] \mid e_2$ ***precedes***(x) e_1 & e_2 ***overlaps(ε)*** $e_3\}$
$\rightarrow p'_{123} \rightarrow d'_{123}$

represents a different problem and therefore, a different diagnosis.

In the case that the above specification designates a given diagnosis and it is determined that e_1 did not follow e_2 after time x, a diagnosis engine issues an anomaly (no concrete diagnosis is derived).

An event has a series of event attributes, which we represent as:

$e = (f_1, f_2, f_3 \ldots, f_n)$
where f: *(value:V)*,
where V is the type of the attribute

Examples of event attributes we consider are:

f_1: *ID*
f_2: *source*
f_3: *timestamp*
f_4: *timezone*
f_5:*English text defining the potential cause*
etc

$e.f_3$ represents the value of attribute f_3 in event e.

The operators on relative event position (**follows**, **overlaps**, etc.) are related to the attributes f_3 and f_4.

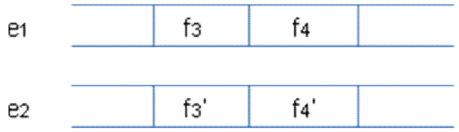


Figure 2.Timestamp and timezone event fields

In this example, $e_1.f_4$ and $e_2.f_4'$ are known, since they represent the timezones of the sources of the two events. Only $e_1.f_3$ and $e_2.f_3'$ need to be set by the local clocks. Let us assume that:

clk_1 sets $e_1.f_3$ and clk_2 sets $e_2.f_3'$,
where *clk* is the local clock of the event source.

$|clk_1\text{-}clk_2| \leq \varepsilon_{12}$,
where ε_{12} is the clock skew between the two local clocks for two domains represented by two semantic clouds [2].

e_2 ***follows***(x) e_1 is computed as follows:

$$(e_1.f_3 + \varepsilon_{12}) + x < e_2.f_3 \text{ (for the same time zone)} \quad (1)$$

For different time zones, this becomes:

$$[(e_1.f_3 + \varepsilon_{12}) \blacksquare Abs(e_1.f_4)] + x < (e_2.f_3) \blacksquare Abs(e_2.f_4), \quad (2)$$

where $\blacksquare$ $Abs(e.f_4)$ represents the operator for normalizing the time between timezones.

Following the same logic, e_2 ***overlaps(ε)*** e_3 for different time zones is computed as follows:

$$|(e_2.f_3) \blacksquare Abs(e_2.f_4) - (e_3.f_3) \blacksquare Abs(e_3.f_4)| < \varepsilon_{23} \quad (3)$$

where
|x| is the absolute value of x
and
ε_{23} represents an acceptable error.

These event-based computations are performed each time a diagnosis is triggered and validated.

In the next section we will use this example in the diagnosis scenario.

IV. USING TEMPORAL FEATURES FOR DIAGNOSIS

This section presents a formal specification of the ontology-based diagnosis, considering temporal relations. Let us assume that the diagnosis engine and the Quality of Diagnosis (QoD) engine introduced in [1] have to trigger the following operations: INTERPRET, APPLY, VALIDATE and MARK.

- Diagnosis engine: INTERPRET events from the system.
- Diagnosis engine: APPLY the diagnosis actions.
- Quality of Diagnosis engine:
  VALIDATE the diagnosis actions.
  and
  MARK successful actions.

The APPLY, VALIDATE and MARK functions were shown in [2]. We reconsider the example with INTERPRET functionality as well.

As discussed in [2], there is a semantic tag hierarchy within each domain, with special dependency relations between semantic tags. Within a domain, semantic tags and their relations form a semantic tag cloud; a domain might have multiple semantic tag clouds associated with it. Let us assume that a system is represented by two semantic tag clouds (Figure 3). Semantic cloud #1 defines the tags and their relationships for a fault related to a power supply while Semantic cloud #2 relates to a potentially real-time and latent fault.

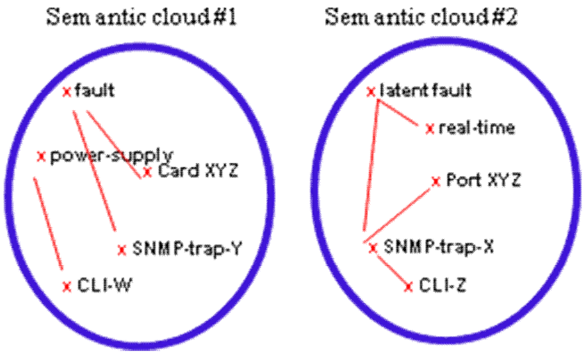


Figure 3. Two Semantic Tag Clouds [2]

When some event patterns occur and diagnosis actions must be triggered (and validated), the Diagnosis Engine interprets the events from the system and applies the diagnosis actions. Next, the Quality of Diagnosis engine validates the actions and marks the successful actions.

The following algorithm is used by the engines to perform the required actions for a given occurrence of combinations of events. A particular series of events occurs as shown in the INTERPRET part of the following algorithm (we use the '.' Notation, i.e., *a.b* means the property '*b*' of the instance '*a*'). When the conditions (2) and (3) explained in Section III hold, the necessary condition to enter the rest of the algorithm is met.

```
START

INTERPRET
IF {[$e_1$, $e_2$, $e_3$] | $e_2$ precedes(x) $e_1$ & $e_2$ overlaps $e_3$}
      CLOCK = $t_0$
      AND $e_1$ belongs to $cloud_1$
      AND $e_2$ belongs to $cloud_2$
      AND $e_3$ belongs to $cloud_2$
      AND $x < t_0$
 THEN
      ERROR
 ELSE
      ASSUME
          $e_2$ precedes(x) $e_1$ & $e_2$ overlaps $e_3$ == TRUE
      AND
IF there is exist $r_c$ < $cloud_1$, $cloud_2$>
     AND $cloud_1$.state = active
     AND $cloud_2$.state = active
     AND
     IF there is $r_{dto}$ <$e_1$, $domain_1$>
        AND $tag_1$ belongs to $domain_1$
        AND $tag_1$ belongs to $cloud_1$
        AND $tag_2$ belongs to $domain_2$
        AND there is $r_T$ <$tag_1$, $tag_2$>
        AND there is $r_{CA1}$ <$cloud_1$, {$action_1$}>
        AND there is $r_{CA2}$ <$cloud_2$, {$action_1$}>
   WITH
        $action_1$ = {$a_1$, $a_3$, $a_6$}
        AND
        $action_2$ = {$a_1$, $a_5$, $a_7$}
THEN

APPLY {{$a_1$, $a_3$, $a_5$, $a_6$, $a_7$} – {
          $a_1$.conflicting ∪
          $a_3$.conflicting ∪
          $a_5$.conflicting ∪
          $a_6$.conflicting ∪
          $a_7$.conflicting}
VALIDATE
          $a_1$.post-conditions = TRUE
          $a_3$.post-conditions = TRUE
          $a_5$.post-conditions = TRUE
          $a_6$.post-conditions = TRUE
          $a_7$.post-conditions = TRUE
MARK
          $a_1$.mode = successful
          $a_3$.mode = successful
          $a_5$.mode = successful
          $a_6$.mode = successful
          $a_7$.mode = successful
END
```

Legend (for details, see [2]):
r_C: R_C | r_C ::= <c_1: C, c_2: C>, cloud to cloud relation
r_T: R_T | r_t ::= <t_1: T, t_2: T>, tag to tag relation
r_{CA}: R_{CA} | r_{CA} ::= <c :C, {a_i : A | p_i: P}>, cloud to action relation
r_{dto}: R_{Dto} | r_{dto} ::= <e:E, d:D>, event to domain relation.

As a result, the successfully marked actions can be re-used as recommended actions when similar event patterns occur. When an event pattern inventory exists, a similar algorithm is associated with each pattern. In this case, the Diagnosis Engine behavior is a combination of all these algorithms.

V. CONCLUSION AND FUTURE WORK

In this article, we proposed a new timestamp approach and considered a series of temporal operators defining event relative temporal position that allows a more fine grain interpretation of system behavior. Based on these concepts, we provided examples on diagnosis interpretations considering temporal dependencies between events and a more complete behavior specification of a diagnosis engine.

As future work, an event dependency pattern repository based on temporal relationships is the target. This will allow a semantic interpretation of different situations and support validations of the actions timely triggered based on probable-cause.

VI. REFERENCES

[1] M. Popescu, P. Lorenz, and J.M. Nicod, An Adaptive Framework for Diagnosis Validation, The Proceedings of The Third International Conference on Advanced Engineering Computing and Applications in Sciences, ADVCOMP 2009, Sliema, Malta, pp. 123-129, IEEE Press

[2] M. Popescu, P. Lorenz, M. Gilg, and J.M. Nicod, Event Management Ontology: Mechanisms and Semantic-driven Ontology, The Proceedigns of The Sixth International confernces on Networking and Servcies, ICNS 2010, Cancun, Mexico, pp. 129 - 136, IEEE Press

[3] W. Stallings, SNMP, SNMPv2, and CMIP: The Practical Guide to Network-Management Standards, Addison-Wesley Publishing Company, 1993, ISBN 0-201-63331-0

[4] M. Popescu et al., US Patent 7275017, Method and apparatus for generating diagnoses of network problems

[5] L. Lamport, TLA: Temporal logic of Actions, http://research.microsoft.com/en-us/um/people/lamport/tla/tla.html

Retrieved: August 12, 2010

[6] R. Griffith, J.L. Helelrstein, G. Kaiser, and Y. Diao, Dynamic Adaptation of Temporal Event Correlation for QoS Management in Distributed Systems, 2006

www.cs.columbia.edu/techreports/cucs-055-05.pdf

Retrieved: August 2, 2010

[7] K. Walzer, T. Breddin, and M. Groch, Relative temporal constraints in the RETE algorithm for complex event detection, Proceedigns of the Second International Conference on Distributed Event-based Systems, 2008, pp. 147-155

[8] M. Popescu, Temporal-oriented policy-driven network management, Master Thesis, Mcgill University, Canada 2000, p. 140

An Algorithm for the Improvement of Tag-based Social Interest Discovery

José Javier Astrain, Alberto Córdoba, Francisco Echarte, Jesús Villadangos
Dept. Ingeniería Matemática e Informática
Universidad Pública de Navarra
Campus de Arrosadía. 31006 Pamplona, Spain
Email: {josej.astrain,alberto.cordoba}@unavarra.es, patxi@eslomas.com, jesusv@unavarra.es

***Abstract*—The success of Web 2.0 has generated many interesting and challenging problems as the discovering of social interests shared by groups of users. The main problem consists on discovering and representing the interest of the users. In this paper, we propose a fuzzy based algorithm that improves the Internet Social Interest Discovery algorithm. This algorithm discovers the common user interests and clusters users and their saved resources by different interest topics. The collaborative nature of social network systems and their flexibility for tagging, produce frequently multiple variations of a same tag. We group syntactic variations of tags using a similarity measure improving the quality of the results provided by the Internet Social Interest Discovery algorithm.**

***Keywords*-Social interest discovering, syntactic variations, collaborative tagging systems**

I. Introduction

Nowadays, one of the problems of social networks in Web 2.0 is the discovering of common interests shared by user communities. Users of a community use to have same interests. In this sense, social communities growth is limited by the definition of scalable and well adapted communities to user interests.

The discovering of social interests shared by groups of users can be focused following three different approaches. The user-centric approach focuses on detecting social interests based on the social connectivity among users [1], [2]. Those works analyse user's social or on-line connections to discover users with particular interests or expertise for a given user. Recent works, as [3], [4] represent the three types of entities that exist in a social tagging system (users, items and tags) by a 3-order tensor, on which latent semantic analysis and dimensionality reduction is performed using both the Higher Order Singular Value Decomposition (HOSVD) method and the Kernel-SVD smoothing technique. In the object-centric approach, [5], [6] explored the common interest among user based on common objects they fetched in peer-to-peer networks. However, without other information of the objects, it cannot differentiate the various social interests on the same object. Furthermore, in Internet social networks such as del.icio.us, most of objects are unpopular. Thus, it is difficult to discover common interest of users on them [7]. The tag-centric [7], [8], [9] approach focuses on directly detecting social interests or topics analysing user annotations. This approach avoids the limitation of object-centric approach [7]. In [10], a tag-centric approach is used to provide semantic resource classification.

Tagging techniques have been widely used in many different social networks. As introduced in [8], the proportion of frequencies of tags within a given site tend to stabilize with time (due to the collaborative tagging by all users). Furthermore, the distribution of frequency of tags for popular sites follows the power law as proved in [11]. This reinforces the need to discover the interests of users since although they use automatic tagging systems, they do so using an uncontrolled vocabulary.

Resource classification can be performed by using clustering techniques using both keywords [12] and tags [7], [10], [13]. An internet social discovery system (ISID) is developed in [7], which cluster users and their saved URLs based on their annotations. Although users may have different interests for an item (and items may have multiple facets) the fact is that tags implicitly describe the users' interests. We discover common interest shared by groups of users in social networks by utilizing user tags. Our approach is based on the insightful study and observation on the user generated tags in social networks systems such as del.icio.us [7]. As users annotate resources, the occurrence of common tags reinforce their common interests.

A same resource can receive different tag annotations from different users; then we consider that a resource has converged when its distribution of tags converges rapidly to a remarkably stable heavy-tailed distribution. Although an increase on the number of annotations also includes an increase on the number of different tags involved in the annotation process, we can observe that most of users agree on the more relevant tags. Those tags are used in a great number of annotations, and by a great number of users. So, a quantification of this agreement degree aids to define a certain threshold in charge of identifying resource convergence. The set of aggregated user tags on a resource is quite compact and stable enough to characterize the same main resource.

One of the main problems of the tag-centric approach is the existence of a high number of syntactic variations (erroneous or not) of other existing tags. A pre-filtering of the tags, as occurs in [14] where the Levenshtein similarity measure is used to reduce the number of tags (identifying syntactic variations), allows increasing the quality of tag clustering minimizing the effects of syntactic variations. In previous works, we proved that the utilization of a fuzzy algorithm (FA_ε) [13] provides best classification results than the obtained when using classical distances as the Levenshtein and Hamming distances; and in [15], we improved those results adding a semantic measure (cosine) to the fuzzy automaton. Cosine similarity, traditionally used in information retrieval [16], measures the similarity between a couple of vectors of n dimensions by finding the cosine of the angle between them. The semantic similarity is obtained comparing the vectorial representation of a couple of terms. In this paper, we present the Fuzzy based Internet Social Discovery algorithm (F_b-ISID), which increases the discovery results provided by ISID [7] by using the fuzzy automaton with ε-moves in conjunction with the cosine similarity to remove the syntactic variations of tags on a folksonomy (*tag cleaning*). The good results obtained show the convenience of re-clustering the tags in order to remove the syntactic variation of tags. The tags containing syntactic variations are clustered in their representative tags preserving their semantic information and reinforcing the tag relevance in the whole set of tags.

The tag cleaning process improves the interest discovering results obtained. Finally, we consider that the appliance of a tag cleaning process must be performed for all the algorithms, which use the related tag-centric approach.

The rest of the paper is organized as follows: Section II describes the F_b-ISID algorithm; Section III describes the experimental results obtained; and finally, conclusions, acknowledgements and bibliographical references end the paper.

II. F_b-ISID DESCRIPTION

In order to deal with the large amount of syntactic variations of tags usually existent in folksonomies, we present the *Fuzzy based-ISID* (F_b-ISID) method. F_b-ISID is based on the pre-filtering of the posts with the aim of increasing the interest discovering results obtained by ISID. For such purpose, F_b-ISID clusters the syntactic variations of tags reducing the entropy of the posts by means of the fuzzy and cosine similarity measures above described. F_b-ISID improves the search of topics of interest against ISID introducing a new component Syntactic Variations which is in charge of the elimination of syntactic variations on tag-centric systems. This section is devoted to describe the components of the F_b-ISID algorithm. The main characteristic of F_b-ISID consists on the introduction of a component, called *SyntacticVariation*, which avoids the syntactic variations of the posts.

The F_b-ISID architecture provides functions as finding topics of interests, resource clustering, and topics of interest indexing.

1) *Finding topics of interest*. For a given set of bookmark post (Bookmark Post is a social bookmarking site alowwing users to submit their blog post and other stories to share with others and make them popular), find all topics of interest. Each topic of interests is a set of tags with the number of their co-occurrences exceeding a given threshold. Those sets of tags, which do not reach this threshold do not give rise to a new topic of interests;
2) *Clustering*. For each topics of interests, find the URLs and the users such that those users have labelled each of the URLs with all the tags in the topic. For each topic, a user cluster and a URL cluster are generated;
3) *Indexing*. Import the topics of interests and their user and URL clusters into an indexing system for application queries.

The components of this architecture are: DATASOURCE, SYNTACTICVARIATION, TOPICDISCOVERY, CLUSTERING and INDEXING.

1) *DataSource*: F_b-ISID inputs are users' posts obtained from social networks as a stream of posts p = *(user,URL, tags)*, where the combination of *user* and *URL* uniquely identifies a post p, and *tags* is the set of tags that the user uses to label the referred URL.
2) *SyntacticVariation*: the discriminator included in this component is in charge of grouping syntactic variations of tags. It computes the fuzzy similarity and the cosine measures among the observed tag and the set of already existing tags (stored in a dictionary) in order to discover syntactic variations of tags. The dictionary includes all the tags that have been used by users in their annotations provided that they are not syntactic variations of other pre-existing tags. The occurrence of a new tag not included in the dictionary implies a clustering process. The identification of a tag as a syntactic variation of an existing tag by the discriminator, implies the assignation of a new tag to the cluster whose cluster-head is the pattern tag with the higher similarity value (pattern). According to the tag lengths, the discriminator calculates the fuzzy similarity or both fuzzy and cosine similarities. Three thresholds Th_1, Th_2 and Th_3, which represent the tag length threshold, the fuzzy similarity threshold and the cosine threshold, respectively, are considered. Whenever the tag length is greater than Th_1, the discriminator uses the fuzzy similarity measure for the tag clustering process. In other case, the cosine measure is also considered by the discriminator in conjunction with the fuzzy similarity measure. If both, fuzzy and cosine measures provided values greater than Th_1 and Th_2 respectively, then the discriminator identifies the tag as a variation of a certain pattern tag, and performs the tag clustering according to this result.

When fuzzy and cosine measures do not agree (values lower than thresholds) the discriminator includes the tag in the dictionary.

3) *TopicDiscovery*: this component is in charge of finding the frequent tag patterns for a given set of post. F$_b$-ISID uses association rules algorithms to identify the frequent tag patterns for the post.
4) *Clustering*: this component collects the posts that contain the tag set (topic), inserting into two collections of clusters (identified by topics) the resources (URLs) and the users of the posts. Its main problem is its complexity, since the algorithm used matches each topic against each post. Then, for a set of n tags, there are 2^n possible topics to check. In order to reduce this complexity, we build a prefix tree over the merged topics. The clustering algorithm for a given set *T* of topics and a given set *P* of post is described in Fig. 1.
5) *Indexing*: this component provides some simple query services for applications:
    - For a given topic, listing all URLs that contain this topic.
    - For a given topic, listing all users that are interested in this topic.
    - For given tags, listing all topics containing the tags.
    - For a given URL, listing all topics that are concerned this URL.
    - For a given URL, and a topic, listing all users that are interested in this topic and have saved the URL.

```
for all topic $t \in T$ do
    $t.user \leftarrow \emptyset$
    $t.url \leftarrow \emptyset$
end for
for all post $p \in P$ do
    for all topic $t$ of $p$ do
        $t.user \leftarrow t.user \cup p.user$
        $t.url \leftarrow t.url \cup p.url$
    end for
end for
```

Figure 1. F$_b$-ISID clustering algorithm.

Figure 2 illustrates the F$_b$-ISID architecture, where the *Syntactic variations* function is added to the ISID architecture. The posts obtained by *DataSource* are processed by *SyntacticVariation*, which clusters the posts avoiding the syntactic variations of tags. The resultant posts (Posts') are then processed by *TopicDiscovery*, which provides the topics of interest. The *Clustering* component clusters these topics and provide the results to the *Indexing* component.

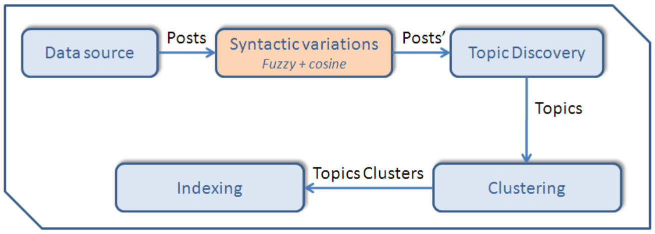


Figure 2. F$_b$-ISID architecture.

In [13] we proposed a method to group syntactic variations of tags using pattern matching techniques. The aim is to cluster in a single centroid all the tags that can be considered as a syntactic variation of a given tag. This centroid represents all the tags included in this cluster, which are syntactic variations of it. In particular, the proposed fuzzy similarity measure (a fuzzy automaton with ε-moves, FA$_\varepsilon$) offers better classification results than other classic techniques (Hamming and Levenshtein measures) after comparing them over a large real dataset. The identification of syntactic variations depends on the length of the tags. Similarity measures perform well for tag lengths equal or greater than five symbols, providing poor results in other cases. In [15] we proposed an hybrid method which adds to the related fuzzy similarity measures a cosine measure in order to improve the clustering process when dealing with short length tags. The use of both cosine and fuzzy similarity measures ensures recognition rates greater than 95% over datasets including large and small length tags. Results have been validated by experts outside the project. By adding the cosine similarity, the tag clustering performed ensures a higher semantic clustering.

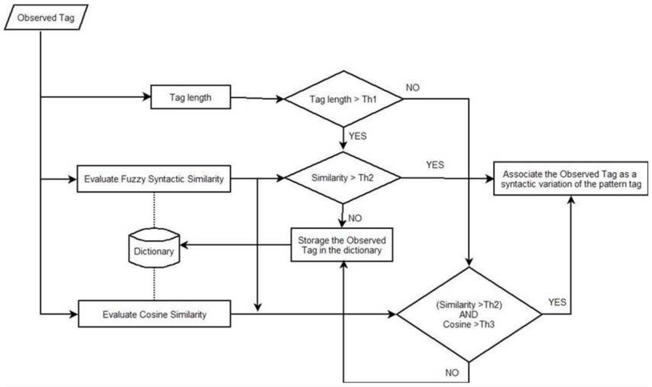

Observed Tag
Tag length
Tag length > Th1
NO
YES
Evaluate Fuzzy Syntactic Similarity
Similarity > Th2
YES
Associate the Observed Tag as a syntactic variation of the pattern tag
NO
Dictionary
Storage the Observed Tag in the dictionary
Evaluate Cosine Similarity
(Similarity >Th2) AND Cosine >Th3
YES
NO

Figure 3. Syntactic variation discovering.

The discriminator used to cluster syntactic variations of tags (see Figure 3) computes the fuzzy similarity and the cosine measures among the observed tag and the set of already existing tags (stored in a dictionary) in order to discover syntactic variations of tags. The occurrence of a new tag not included in the dictionary implies a clustering process. The identification of a tag as a syntactic variation of an existing tag by the discriminator, implies the assignation of a new tag to the cluster whose cluster-head is the pattern tag with the higher similarity value (pattern). The discriminator uses the fuzzy similarity or the fuzzy and cosine similarities according to the tag length. Three thresholds Th_1, Th_2 and Th_3, which represent the tag length threshold, the fuzzy similarity threshold and the cosine threshold, respectively, are considered. Whenever the tag length is greater than Th_1, the discriminator uses the fuzzy similarity measure for the tag clustering process. In other case, the cosine measure is also considered by the discriminator in conjunction with the fuzzy similarity measure. If both, fuzzy and cosine measures provided values greater than Th_1 and Th_2 respectively, then the discriminator identifies the tag as a variation of a certain pattern tag, and performs the tag clustering according to this result. When fuzzy and cosine measures do not agree (values lower than thresholds) the discriminator includes the tag in the dictionary.

III. Experimental results

The comparison between F$_b$-ISID and ISID is performed implementing all the components of both algorithms following the same process and the same presentation of the results than those of [7].

For such purpose, we have retrieved web pages annotated by users from Del.icio.us using its *Recent Bookmarks* page. We consider those resources bookmarked by at least 250 users, storing the URL of those resources. Information has been retrieved during the period from 1-15 March, 2010. A total amount of 419,891 resources, with 2,296,300 annotations, 197,148 users and 156,897 tags have been obtained. We have randomly generated some subsets of posts from a total amount of 779,674 posts. Table I shows the number of tags, users and resources (URLs) for each subset of posts. For example, the subset containing 100,000 posts refers to 36,603 different tags, 44,051 different users and 72,567 different resources. Repetitions are not considered.

TABLE I
Number of different tags, users and URLs for each subset of posts.

Posts	Tags	Users	URLs
5,000	4,546	3,060	4,545
50,000	22,111	23,659	38,984
100,000	36,603	44,051	72,567
200,000	59,424	74,502	134,424
300,000	59,398	99,144	192,462
400,000	97,430	121,498	245,929
500,000	114,358	146,972	293,685
600,000	130,451	166,546	339,875
700,000	145,372	183,542	384,663
779,674	156,897	197,148	419,891

The execution of the *SyntacticVariation* component over the dataset retrieves a total amount of 991 syntactic variations (4.31% of the 779,674 posts) with a recognition rate of the 96.97%, which has been verified manually with the aid of Wordnet and Wikipedia. Table II shows the distribution of the syntactic variations of tags.

TABLE II
Syntactic variations distribution.

	Occurrence distribution	
Syntactic variation	Number	Percentage
Number (singular/plural)	772	77,90
Delimiters	109	10,99
Synonyms	59	5,95
Misclassification	30	3,03
Other	21	2,12
Total	991	100

Table III shows the number of subsets generated by both algorithms. It can be seen that the number of tag subsets generated by F$_b$-ISID is lower than the number of tag subsets generated by ISID. We observe how F$_b$-ISID (347,985,324) obtains a 24.10% less of subset tags than ISID (458,452,178). The suppression of syntactic tag variations causes the clustering of those concepts scattered in many different terms. That allows the apparition of new topics of interest since the new subset of tags reach the threshold required to become a topic of interest.

TABLE III
Number of tag subsets.

	Number of subsets		Variation
Posts	ISID	F$_b$-ISID	(%)
5,000	6,590,055	5,602,342	14,99
50,000	60,905,524	50,694,270	16,77
100,000	113,015,128	95,877,191	15,16
200,000	203,577,716	173,393,490	14,83
300,000	283,734,103	245,137,336	13,60
400,000	338,641,100	256,157,047	24,36
500,000	340,920,037	309,822,299	9,12
600,000	408,342,102	323,623,856	20,75
700,000	437,792,108	336,625,280	23,11
779,674	458,452,178	347,985,324	24,10

A comparative study between F$_b$-ISID and ISID is presented in Table IV, which shows the results obtained for the grouping of contents: Topics & Users, Topics & URLs, Users & URLs, Topics of interests, Users and URLs.

1) Topics & Users: F$_b$-ISID improves the classification a 11.16%.
2) Topics & URLs: F$_b$-ISID improves the classification a 11.40%
3) Topics of interests: F$_b$-ISID improves the classification a 20.15%
4) Users & URLs: similar results for both F$_b$-ISID and ISID (0.04%).
5) Users: similar results for both F$_b$-ISID and ISID (0.07%).
6) URLs: similar results for both F$_b$-ISID and ISID (0.06%).

TABLE IV
GROUPING: TOPICS, USERS AND URLS FOR EACH SUBSET OF POSTS.

Number	ISID					
of posts	Topics & Users	Topics & URLs	Users & URLs	Topics	Users	URLs
5,000	4,517	5,562	3,034	92	1,959	2,726
50,000	98,334	107,774	39,375	1,250	19,558	30,167
100,000	261,435	268,875	82,166	3,140	37,915	58,671
200,000	637,717	632,257	170,052	7,331	66,314	112,560
300,000	1,054,851	1,042,349	258,396	11,277	89,478	163,302
400,000	1,498,064	1,447,532	347,008	14,574	110,577	210,256
500,000	1,914,231	1,779,109	433,676	14,836	133,972	250,608
600,000	2,378,987	1,856,432	554,765	16,786	145,786	287,654
700,000	2,944,428	2,624,802	612,216	20,567	169,052	330,955
779,674	3,014,563	2,765,498	686,056	22,012	172,987	348,765

Number	F_b-ISID					
of posts	Topics & Users	Topics & URLs	Users & URLs	Topics	Users	URLs
5,000	4,822	5,890	3,108	100	2,001	2,782
50,000	110,045	119,629	39,579	1,393	19,642	30,323
100,000	294,191	298,999	82,202	3,483	37,923	58,697
200,000	731,529	714,237	170,089	8,402	66,313	112,589
300,000	1,219,574	1,188,299	258,740	13,243	89,538	163,566
400,000	1,574,956	1,652,038	345,351	14,701	110,269	209,100
500,000	2,262,048	2,059,656	434,885	18,735	134,220	251,424
600,000	2,528,314	2,197,279	559,234	24,765	145,876	288.765
700,000	2,972,815	2,630,037	604,828	15,837	169,806	331,045
779,674	3,393,105	3,121,478	686,056	27,566	173,102	348,964

TABLE V
INDEXING: ”BLOG” AND ”HTTP://ANIMOTO.COM”.

Number	ISID					F_b-ISID				
of		blog		animoto			blog		animoto	
posts	URLs	Users	Topics	Topics	Users	URLs	Users	Topics	Topics	Users
5,000	124	106	1	5	0	164	137	1	7	0
50,000	1,213	920	26	53	1	1,513	1,200	46	57	2
100,000	2,244	1,818	60	88	5	2,845	2,361	107	96	7
200,000	4,163	3,344	159	232	8	5,301	4,329	274	251	14
300,000	6,189	4,866	257	297	14	7,761	6,235	463	312	25
400,000	8,042	6,357	361	377	27	10,078	8,134	513	351	40
500,000	9,848	8,015	364	472	32	12,329	10,248	693	498	49
600,000	11,461	9,467	424	502	35	13,564	12,087	484	521	55
700,000	13,257	10,937	504	593	44	16,539	14,363	581	613	63
779,674	14,573	12,146	564	626	50	18,323	15,315	629	643	71

One can note that F_b-ISID provides best classification results when grouping results by topic, while classification results remain unchanged when only considering User and URL clustering.

The topic *blog* and the URL *http://animoto.com* are used to build the following basic queries:

1) For the topic *blog*, list all the URLs associated with the tag *blog*.
2) For the topic *blog*, list all users that are interested in this topic.
3) For tag *blog*, list all topics containing the tag *blog*.
4) For the URL *http://animoto.com*, list all the topics containing the resource *animoto*.
5) For the URL *http://animoto.com* and the topic *blog*, list all the users interested in the topic *blog* that have saved the URL *animoto*.

Table V shows the results obtained for the subsets of posts, namely URLs, Users and Topics for *blog*, and Topics and Users for *http://animoto.com*. The results obtained show that F_b-ISID:

a) obtains a 25,73% of URLs containing the topic *blog*;
b) increases the number of users interested in *blog* (29,05%);
c) increases the number of topics containing *blog* (11,52%);
d) increases the number of topics that are related to *http://animoto.com* (4,10%);
e) increases the number of users interested in *blog*, which use *http://animoto.com* (23,44%).

Fig. 4 shows that F_b-ISID provides best results for the number of URLs and Users related with the topic *blog* for each of the post sets.

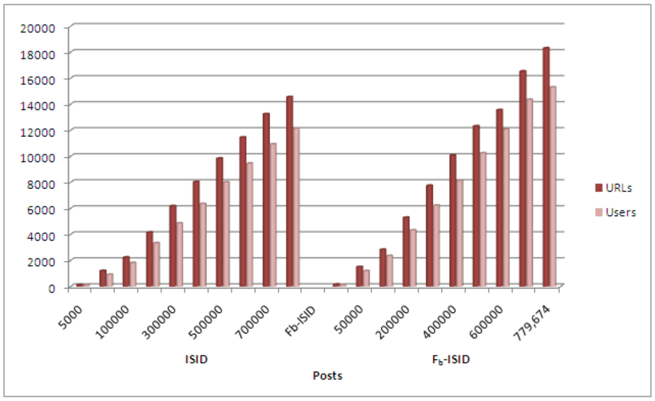


Figure 4. Comparative of URLs and users related with the topic blog.

IV. Conclusions

In this paper, we have introduced a new algorithm, named F_b-ISID, for the discovering of the interest of users in collaborative tag-based systems. The experiments performed show that F_b-ISID obtains better results than the ISID algorithm. This good behaviour is due to the fact that the architecture of F_b-ISID contains one component not included in ISID. This component filters or groups together syntactic variations of the tags contained in the initial posts. In this way, F_b-ISID obtains more topics of interests and performs basic queries more efficiently than ISID. Finally, we consider that the clustering of syntactic variation in the data sources of social systems, improves the performance of algorithms for interests discovery, based on tag-centric approaches.

Acknowledgments

Research partially supported by the Spanish Research Council under research grants TIN2008-03687 and TIMI-2006-CENIT-4834.

References

[1] M. F. Schwartz and D. C. M. Wood, "Discovering shared interests using graph analysis," *Commun. ACM*, vol. 36, no. 8, pp. 78–89, 1993.

[2] N. Ali-Hasan and L. A. Adamic, "Expressing social relationships on the blog through links and comments," in *International Conference on Weblogs and Social Media (ICWSM)*, 2007. [Online]. Available: http://www.icwsm.org/papers/2–Ali-Hasan–Adamic.pdf

[3] P. Symeonidis, "User recommendations based on tensor dimensionality reduction," in *Artificial Intelligence Applications and Innovations III*, ser. IFIP Advances in Information and Communication Technology. Springer Boston, 2009, vol. 296, no. 296, pp. 331–340. [Online]. Available: http://dx.doi.org/10.1007/978-1-4419-0221-4-39

[4] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis," *IEEE Transactions on Knowledge and Data Engineering*, no. To appear, 2010.

[5] K. Sripanidkulchai, B. M. Maggs, and H. Zhang, "Efficient content location using interest-based locality in peer-to-peer systems," in *INFOCOM'2003: 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*. IEEE, march 2003, pp. 2166–2176.

[6] L. Guo, S. Jiang, L. Xiao, and X. Zhang, "Fast and low-cost search schemes by exploiting localities in p2p networks," *J. Parallel Distrib. Comput.*, vol. 65, no. 6, pp. 729–742, 2005.

[7] X. Li, L. Guo, and Y. E. Zhao, "Tag-based social interest discovery," in *WWW '08: Proceeding of the 17th international conference on World Wide Web*. New York, NY, USA: ACM, 2008, pp. 675–684.

[8] S. A. Golder and B. A. Huberman, "Usage patterns of collaborative tagging systems," *J. Inf. Sci.*, vol. 32, no. 2, pp. 198–208, April 2006. [Online]. Available: http://dx.doi.org/10.1177/0165551506062337

[9] Z. Yin, R. Li, Q. Mei, and J. Han, "Exploring social tagging graph for web object classification," in *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2009, pp. 957–966.

[10] F. Echarte, J. J. Astrain, A. Córdoba, J. Villadangos, and A. Labat, "Acoar: a method for the automatic classification of annotated resources," in *K-CAP '09: Proceedings of the fifth international conference on Knowledge capture*. New York, NY, USA: ACM, 2009, pp. 181–182.

[11] H. Halpin, V. Robu, and H. Shepherd, "The complex dynamics of collaborative tagging," in *WWW '07: Proceedings of the 16th international conference on World Wide Web*. New York, NY, USA: ACM, 2007, pp. 211–220.

[12] C. H. Brooks and N. Montanez, "Improved annotation of the blogosphere via autotagging and hierarchical clustering," in *WWW '06: Proceedings of the 15th international conference on World Wide Web*. New York, NY, USA: ACM, 2006, pp. 625–632.

[13] F. Echarte, J. J. Astrain, A. Córdoba, and J. Villadangos, "Improving folksonomies quality by syntactic tag variations grouping," in *SAC '09: Proceedings of the 2009 ACM symposium on Applied Computing*. New York, NY, USA: ACM, 2009, pp. 1226–1230.

[14] L. Specia and E. Motta, "Integrating folksonomies with the semantic web," in *ESWC '07: Proceedings of the 4th European conference on The Semantic Web*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 624–639.

[15] J. J. Astrain, F. Echarte, A. Córdoba, and J. Villadangos, "A tag clustering method to deal with syntactic variations on collaborative social networks," in *ICWE'09: Proceedings of the 9th International Conference on Web Engineering*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 434–441.

[16] A. G. Maguitman, F. Menczer, H. Roinestad, and A. Vespignani, "Algorithmic detection of semantic similarity," in *WWW'2005: Proceedings of the 14th international conference on World Wide Web*. New York, NY, USA: ACM, 2005, pp. 107–116.

Search and Navigation in Semantically Integrated Document Collections

Saša Nešić, Fabio Crestani, Mehdi Jazayeri
Faculty of Informatics,
University of Lugano
Lugano, Switzerland
{*sasa.nesic, fabio.crestani, mehdi.jazayeri*}*@usi.ch*

Dragan Gašević
School of Computing and Information System,
Athabasca University
Athabasca, Canada
dgasevic@acm.org

***Abstract*—The paper presents a novel approach to semantic search and navigation in office-like document collections. The approach is based on a semantic document model that we have developed to enable unique identification, semantic annotation, and semantic linking of document units of office-like documents. In order to semantically annotate document units and to link semantically related document units, we first conceptualize document units' semantics and represent them by vectors of ontological concepts and their corresponding weight vectors. In the semantic search, we represent a user query by a query's concept vector, which is generated in the same way as document units' concept vectors, and then determine the search results by measuring the similarity between the query's and the document units' concept weight vectors. After the search, by following the semantic links of a selected document unit, the user can navigate through the document collection and discover semantically related document units. Results of the preliminary evaluation, conducted with a prototype implementation, are promising. We present a brief analysis of these results.**

***Keywords*-semantic search, semantic linking and navigation;**

I. Introduction

During the last decade a considerable number of ontology-driven information retrieval approaches [1], [2], [3], [4] has been developed to enhance the search and retrieval by making use of available semantic annotations and their underlining ontologies. Central to the ontology-driven information retrieval is the problem of having substantial amount of accurate semantic annotations. Most existing semantic annotation approaches [5] are based on the syntactic matching of ontological concept descriptions against document content. In spite of advanced data mining and NLP techniques applied in these approaches, usually poor and ambiguous concept descriptions lead to insufficient and inaccurate semantic annotation. Few approaches, such as [2], try to enhance the semantic annotation by extending the set of syntactic matches with related concepts from the ontology, discovered by utilizing formal ontological semantics. Such concepts are usually referred to as semantic matches. The combination of the syntactic and semantic matching can increase the amount of semantic annotations, but it opens the problem of the concept relevance [6]. Therefore, one of the most important issues in this scenario is how to assess the relevance weight of the discovered semantic matches and to use only the most relevant of them.

In this paper we present a unified solution that should enable efficient semantic search and navigation in document collections holding semantically related data. The solution is based on the novel document representation model, namely semantic document model (SDM) [7] which comprises the publishing document data in RDF, the semantic annotation and indexing of document data by weighted ontological annotations and the semantic linking of related data within the document collections. By the weighted annotations, which we calculate based on the semantic distances between concepts in the annotation ontology, we intend to improve the semantic search in document collections. By the publishing document data in RDF and the semantic linking of document data, we intend to enable client applications to easily navigate between documents and to discover semantically related data.

The rest of the paper is organized as follows. In Section II we outline main characteristics of the SDM model. In Section III we describe our approach to concepts discovery in document units, especially focusing on a novel, concept exploration algorithm that we apply in the semantic matching. Section IV explains the way we use the discovered concepts for the semantic annotation, indexing and linking of document units of a given document collection. In Section V we present the semantic search and navigation services, which utilize the semantic annotations and links to search and navigate in semantically integrated document collections. In Section VI we discuss the results of the preliminary evaluation that we conducted as a proof of concept. We conclude the paper with Section VII, giving some final remarks and discussing our plans for the future work.

II. Semantic Document Model

We have created a novel document representation model, namely semantic document model (SDM) [7], aiming to provide the infrastructure for the unique identification, the semantic annotation and the semantic linking of fine-grained units of document data. SDM represents document data as RDF [8] linked data, providing an RDF node for each document unit. Document units are uniquely identified by the means of HTTP dereferencable URI of their RDF nodes, semantically annotated by ontological concepts from domain ontologies, and linked to other document units by RDF links that model hierarchical, structural and semantic relationships among them. SDM is formally described by the *smd ontology* [7], which specifies possible types of document units (e.g., `sdm:paragraph`, `sdm:section`, `sdm:table` and `sdm:illustration`), types of hierarchical and structural relationships among document units (e.g., `sdm:hasPart`, `sdm:isPartOf` and `sdm:belongsTo`), and the semantic annotation and the semantic linking interfaces.

The semantic annotation interface consists of the `sdm:Annotation` entity with its two properties: the `sdm:annotationConcept` property that holds a reference to the concept from an annotation ontology and the `sdm:conceptWeight` property that determines the relevance of the annotation concept for the document unit it annotates. The semantic linking interface consists of the `sdm:SemanticLink` entity and the following properties: the `sdm:unitOne` and `sdm:unitTwo`, which hold the document units to be linked, the `sdm:relationshipConcept` property that holds the reference to the ontological concept that annotates both units and determines the type of the semantic relationship, and the `sdm:linkStrength` property that determines the strength of the semantic relationship between the document units. As we can see from the specification of the semantic annotation and the semantic linking interfaces, both of them require the concepts from domain ontologies that conceptualize human-readable information stored in document units. Therefore, the concept discovery represents the foundation of the semantic annotation and linking in SDM.

III. Concepts Discovery in Document Units

The concept discovery that we propose, combines the syntactic matching of lexically expanded concept descriptions with the semantic matching by applying the concept exploration algorithm. In the rest of the section we first describe the main characteristics of the proposed syntactic and the semantic matching and then give detailed description of the concept exploration algorithm.

A. Syntactic and Semantic Matching

Any domain ontology can be represented as a graph $O := (\mathbb{C}, \mathbb{R}, H^C, H^R)$ where $\mathbb{C} = \{c_1, c_2, c_3, ..., c_n\}$ is a set of concepts, $\mathbb{R} = \{r_1, r_2, ..., r_m\}$ is a set of relations and H^C, H^R are hierarchies defining a partial order over concepts and relations respectively. Moreover, each concept is described with a set of labels. For example, the set of labels of the concept c_i is $\mathbb{L}_i = \{l_{i1}, l_{i2}, ..., l_{im}\}$. In practice, however, ontology engineers provide only one label for each ontology concept or even neglect to label concepts considering human readable parts of concept URIs to be concept labels [5]. In order to cope with this problem, which can lead to inefficient syntactic matching, prior to the syntactic matching we perform the lexical expansion of the concept descriptions with related terms from lexical dictionaries such as WordNet [9].

The objective of the syntactic matching is to analyze the content of a document unit (*DU*) and to check if some of the concept labels, appear in it. For the concepts whose labels appear in the *DU*, we calculate the concept weight by taking into account the following: 1) the labels' origin factor that makes distinction between original concept labels and those from the lexical expansion, 2) the labels' frequency of occurrence in the *DU* and 3) the inverse *DU*' frequency in a document collection. The result of the syntactic matching is the concept vector of the *DU* and the corresponding concept weight vector. For example, if we have a document unit d that is being annotated, after the syntactic matching we got the following concept vector: $\overrightarrow{d} = [c_1, c_2, ..., c_r]$; $c_i \in \mathbb{C}$ and the corresponding concept weight vector $\overrightarrow{W_C}(d) = [w_{c_1}, w_{c_2}, ..., w_{c_r}]$, where w_{c_i} is the relevance weight of the concept c_i for the document unit d.

The objective of the semantic matching is to extend the concept vector $\overrightarrow{d}$, which is formed as a result of the syntactic matching, with semantically related concepts from the annotation ontology. By applying the concept exploration algorithm, which we explain in detail in the following section, to each of the document unit's syntactic matches, we discover the document unit's semantic matches and form the expanded concept vector $\overrightarrow{d}^e = [c_1, c_2, ..., c_r, c_{e1}, ..., c_{em}]$. For each of the semantic matches c_{ej} the algorithm calculates the semantic distance $SDist^c(c_{ej}, c_i)$ from the initial syntactic match $c_i \in \overrightarrow{d}$. The weight $w_{c_{ej}}$ of the semantic match c_{ej} for the document unit d is then calculated by the following formula:

$$w_{c_{ej}} = w_{c_i} * \beta^{-SDist^c(c_{ej}, c_i)}; \quad \beta > 1 \qquad (1)$$

where w_{c_i} is the weight of the syntactic match c_i and β is a generic coefficient. We devised the formula (1) so that it satisfies boundary conditions regardless of the value of coefficient β. For the first boundary condition $SDist^c(c_{ej}, c_i) = 0$, meaning that the concepts c_{ej} and c_i are semantically identical, $w_{c_{ej}} = w_{c_i}$, that is, the weight of the semantic match is the same as the weight of the initial syntactic match. For the second boundary condition $SDist^c(c_{ej}, c_i) \to \infty$, meaning that the concepts c_{ej} and c_i are semantically unrelated, $w_{c_{ei}} \to 0$, that is, the weight of the semantic match tends towards zero. For $SDist^c(c_{ej}, c_i) \in (0, \infty)$, the optimal value of coefficient β has to be experimentally determined. For the evaluation, which results we discuss in Section VI, we used the exponential constant e as the value of coefficient β thus making (1) belongs to the family of negative exponential functions.

B. Concept Exploration Algorithm

The main assumption on which the concept exploration algorithm runs is the possibility to associate numerical values to ontological relations in the annotation ontology and to form the weighted ontology graph. We refer to these values as *the relation semantic distances ($SDist^r$)*. Moreover, we distinguish between two types of the relation semantic distance: 1) $SDist^r_{\mathcal{D}\rightarrow\mathcal{R}}(r)$ determining semantic distance of the concepts belonging to the domain ($\mathcal{D}$) of the relation r from the concepts belonging to the range ($\mathcal{R}$) of r, and 2) $SDist^r_{\mathcal{R}\rightarrow\mathcal{D}}(r)$ determining the semantic distance of the concepts belonging to the range of r from the concepts belonging to the domain of r. In general, the values of the relational semantic distances can be: 1) specified at design time of the ontology by the domain experts, 2) experimentally devised by using a controlled knowledge/data base and 3) learned over time by exploiting the ontology in real world applications within the ontology domain. Based on our experience the choice between these three strategies is strongly domain-dependent. A combination of the strategies is also valid.

The general idea of the algorithm (see Algorithm 1) is to explore the ontology graph starting from the input concept to find all concepts which satisfy the given semantic distance constraint (SD_c) and the given path length constraint (PL_c). SD_c is the maximum allowed semantic distance between the input and target concepts. PL_c is the maximum number of hops (i.e., ontology relations) allowed to belong to a path between the input and target concepts. The algorithm takes the following input: the weighted ontology graph O_w formed by associating values of the relation semantic distances to the ontology relations, the input concept c, the semantic distance constraint SD_c, and the path length constraint PL_c. The output consists of a vector of discovered related concepts $\overrightarrow{C_e}$ and a vector of the semantic distances $\overrightarrow{SD_e}$ between the discovered concepts and the input concept. The algorithm starts by the $Paths1(O_w, c, PL_c)$ function (line 3) which constructs a set of all possible acyclic paths $\mathbb{P}$, starting from the input concept c and whose length is less than PL_c. Next, (line 4) the $Concepts(\mathbb{P})$ function extracts all concepts from the set of paths $\mathbb{P}$ and forms a distinct set of extracted concepts $\mathbb{C}$. Next, (line 6) for each concept $c_i \in \mathbb{C}$ function $Paths2(c, c_i, \mathbb{P})$ returns a set of paths $\mathbb{P}_i$ ($\mathbb{P}_i \subseteq \mathbb{P}$) which start in concept c and end in concept c_i. Next, (line 8) for each path $p_{ij} \in \mathbb{P}_i$ between c and c_i, function $SDist^p(p_{ij})$ calculates the semantic distance of the path that we refer to as *the path semantic distance ($SDist^p$)*. The function actually sums the relation semantic distances of relations that make the path. For those relations $r_k \in p_{ij}$ with the same direction as a direction $c \rightarrow c_i$, the function takes $SDist^r_{\mathcal{R}\rightarrow\mathcal{D}}(r_k)$ while for r_k with the direction $c \leftarrow c_i$, the function takes $SDist^r_{\mathcal{D}\rightarrow\mathcal{R}}(r_k)$.

Algorithm 1 Concept Exploration Algorithm

```
1: INPUT $O_w, c, \overrightarrow{SD_c}, PL_c$
2: OUTPUT $\overrightarrow{C_e}$, $\overrightarrow{SD_e}$
3: $\mathbb{P} = Paths1(O_w, c, PL_c) = \{p_1, ..., p_m\}$ {finds all paths from $c$ with a length $\leq PL_c$}
4: $\mathbb{C} = Concepts(\mathbb{P}) = \{c_1, ..., c_n\}$ {extracts all concepts from the set of paths $\mathbb{P}$}
5: for all $c_i$ such that $c_i \in \mathbb{C}$ do
6:     $\mathbb{P}_i = Paths2(c, c_i, \mathbb{P}) = \{p_{i1}, ..., p_{ik}\}$ {finds a set of acyclic paths $\mathbb{P}_i \subset \mathbb{P}$ between $c$ to $c_i$}
7:     for all $p_{ij}$ such that $p_{ij} \in \mathbb{P}_i$ do
8:         $SDist^p(p_{ij})$ {calculates the semantic distance of path $p_{ij}$}
9:     end for
10:    $SDist^c(c_i, c)$ {calculates the semantic distance of the concept $c_i$ from $c$}
11: end for
12: $\overrightarrow{C_e} = [c_1, ..., c_p]$, $c_i \in \mathbb{C}$ and $SDist^c(c_i, c) \leq SD_c$
13: $\overrightarrow{SD_e} = [SDist^c(c_1, c), .., SDist^c(c_p, c)]$
```

After the algorithm calculates the path semantic distances of all paths $\mathbb{P}_i$, it calculates the semantic distance of concept c_i from the input concept c by applying function (2). We call this semantic distance *the concept semantic distance* ($SDist^c$). $SDist^c(c_i, c)$ can be also considered as the relation semantic distance $SDist^r_{\mathcal{R}\rightarrow\mathcal{D}}(r(c, c_i))$ of a new single relation $r(c, c_i)$ from the concept c to c_i.

$$SDist^c(c_i, c) = SDist^r_{\mathcal{R}\rightarrow\mathcal{D}}(r(c, c_i)) = \frac{1}{\sum\limits_{j=1}^{k} \frac{1}{SDist^p(P_{ij})}} \tag{2}$$

We designed function (2) so that it prioritizes the impact of those paths with the small path semantic distance, in determining the concept semantic distance. Finally, the algorithm discards all concepts from the set $\mathbb{C}$ which do not satisfy the SD_c constraint, and forms the output vector of the discovered related concepts $\overrightarrow{C_e}$, along with the vector of their semantic distances $\overrightarrow{SD_e}$ from the input concept c.

IV. Semantic Documents Annotation, Linking and Indexing

The semantic annotation of document units defined by SDM refers to the process of linking the discovered ontological concepts and their weights to document units' RDF nodes, via the SDM annotation interface. If documents of a given document collection are annotated by concepts from the same, shared domain ontology, then implicit semantic relationships between their document units can easily be identified and made explicit. For example, if two document units are annotated by the same ontological concept, it means that they share some semantics and there is an implicit semantic relationship between them. By setting up explicit

RDF links between RDF nodes of document units, based on the SDM linking interface, we bring the collection's data into an integrated information space. Following the semantic links, the user can easily navigate in this information space and discover semantically related data. Moreover, by exposing the SPARQL HTTP endpoints [8] to RDF repositories of the document collections, we can enable the integration of distant document collections. In this way, the user can navigate through semantically related data belonging to different document collections as well.

Finally, besides the semantic annotation and linking, the discovered concepts are also used for the concept indexing of the document units. The concept index contains a list of concepts (i.e., concept identifiers) from the annotation ontologies, each of which is assigned a list of the document units it annotates. For each document unit in the concept's list, the index also stores the weight of the concept for the document unit. The concept index plays the key role in the semantic search that we discuss in the next section.

V. Semantic Search and Navigation

In this section we describe the semantic search and the semantic navigation services, which we have developed as parts of a broader, service oriented architecture called the Semantic Document Architecture - SDArch [10]. These two services provide the mechanisms for the semantic search and navigation in the collections of documents represented by SDM. In order to provide the user interface to the SDArch services, we have developed a set of tools called 'SemanticDoc' and integrated them into MS Office as add-ins. Further information about the SDArch services and SemanticDoc tools can be found on our project web page [11].

The search process normally starts with the user constructing a query that reflects her information needs. As the initial form of the user query, the semantic search service takes a free text query. The service then models the semantic meaning of the query by forming a weighted query concept vector, which we refer to as a semantic query. The search service actually applies the syntactic and the semantic matching to discover ontological concepts, which conceptualize the semantic meaning of the query. For each discovered concept, the service calculates its relevance weight to the query and forms a weighted query concept vector. The way the search service forms semantic queries is quite similar to the process of the concepts discovery (Section III) in document units.

Having both the document units and the user query represented in the same way, by their weighted concept vectors, the rest of the search process proceeds as follows. From the concept index of the selected document collection, the search service discovers document units which are indexed by the concepts from the semantic query. Then, the service calculates the similarity between the discovered document units and the semantic query, by computing the similarity between the document units' concept vectors and the query's concept vector. At the end, the service ranks the document units based on the calculated similarity and retrieves the ranked list of the document units.

The semantic navigation service enables users to traverse document collections by navigating along the semantic links. The navigation process assumes the existence of an exploratory interface through which the users interact with the semantic navigation service (i.e., SemanticDoc [11] tools). The navigation process starts by the user browsing the initial document unit and clicking on one of the unit's annotation concepts (i.e., concept label). This activates the navigation service, which takes the URIs of the document unit and the concept, forms a SPARQL query (see Fig. 1), and executes it against the collection's RDF repository. Since the initial document unit can be linked to many document units via the same semantic link, thus the query can return multiple document units. The query orders the return document units by the strength of the semantic links between them and the initial document unit. After the query execution, the navigation service sends the list of the document units to the browse in which the user browses their details.

```
PREFIX sdm: <http://www.semanticdoc.org/sdm.owl#>
SELECT ?targetUnit ?strenght
WHERE{?link sdm:relationshipConcept concept_a32c154
      ?link sdm:unitOne unit_b42c177
      ?link sdm:unitTwo ?targetUnit
      ?link sdm:linkStrength ?strength }
ORDER BY ?strength
```

Figure 1. Example of a SPARQL query executed by the navigation service

VI. Evaluation

In order to evaluate the usability of the semantic search and navigation services we conducted a usability study in which we considered the user effectiveness, efficiency and satisfaction in using the services, while preparing a course material. Both, the quantitative measures (e.g., task execution time, number of window switches and number of mouse clicks) and the users' subjective feedback, collected by the evaluation questionnaire and the series of interviews, showed positive results. The detailed discussion on the usability study can be found in [12]. In this paper, however, our focus is on the experimental evaluation of the semantic annotation (i.e., the concept discovery) and the semantic search.

The experimental evaluation that we discuss hereafter, was designed more as a proof of concept; it was not meant to address issues of scalability or efficiency. The document collection that we used in the experiments was composed of 170 Word documents (2735 paragraphs - document units of interest for these experiments) containing records for steel, aluminum, copper, titanium, and other metals. We optioned the collection from KEY-to-METALS [13] company, which

Semantic relation	Representation	$SDist^r_{\mathcal{R}\to\mathcal{D}}(r)$	$SDist^r_{\mathcal{D}\to\mathcal{R}}(r)$
hypernym	$skos : broader$	$1-\delta_{hyper}=0,53$	$1-\delta_{hypo}=0,16$
hyponym	$skos : narrower$	$1-\delta_{hypo}=0,16$	$1-\delta_{hyper}=0,53$
holonym	$skos : relatedPartOf$	$1-\delta_{holo}=0,88$	$1-\delta_{mero}=0,84$
meronym	$skos : relatedHasPart$	$1-\delta_{mero}=0,84$	$1-\delta_{holo}=0,88$
synonym	$owl : equivalentClass$	$1-\delta_{syn}=0,30$	$1-\delta_{syn}=0,30$
identical	$owl : sameAs$	0	0

Table I
RELATION SEMANTIC DISTANCES IN METALS ONTOLOGY

Strategy	Number of concepts	Number of syn. matches	Number of sem. matches	Avg. weight of syn. match.	Avg. weight of sem. match.
S_1	211	1524	-	2.56	-
S_2	343	3182	-	3.62	-
S_3	672	3182	6714	3.62	2.43
S_4	795	3182	11102	3.62	1.12
S_5	924	3182	23716	3.62	0.27

Table II
CONCEPT DISCOVERY RESULTS FOR STRATEGIES (S_1-S_5)

maintains one of the world's most comprehensive metals database. As the annotation ontology we used the *Metals* ontology, which we also got from the same company. The ontology contains over $3,500$ concepts about metals and their applications. It is an OWL ontology which conforms to the SKOS specification [6]. SKOS defines a family of relations such as $skos : narrower$, $skos : broader$ and $skos : related$ for expressing simple relationships between concepts within an ontology.

Table I shows a subset of semantic relations in the *Metals* ontology, along with their SKOS and OWL representations and values of the relation semantic distances. The values of the relation semantic distances were assessed based on the results of the experimental studies [14]. In these studies the authors measured the semantic similarity/relatedness between terms in WordNet, connected via the *hypernymy*, *hyponymy*, *holonymy*, *meronymy* and *synonymy* relations, and produced the following values: $\delta_{hyper} = 0.47$, $\delta_{hypo} = 0.84$, $\delta_{holo} = 0.12$, $\delta_{mero} = 0.16$ and $\delta_{syn} = 0.70$. Value $\delta_r = 0$ means that two terms are semantically unrelated via relation r, and $\delta_r = 1$ that the terms are semantically identical. We calculate the values of the relation semantic distances as $1 - \delta_r$ and take into account the fact that *hypernymy* and *hyponymy* as well as *holonymy* and *meronymy* are mutually inverse relations. Moreover, the *Metals* ontology contains the $owl : sameAs$ relation which links two semantically identical concepts/individuals, so that both of the relation semantic distances have been assessed as zero.

In order to evaluate the semantic annotation, we have transformed the document collection by applying five different concept discovery strategies: S_1 - simple syntactic matching, S_2 - lexically expanded syntactic matching, and S_3, S_4, S_5 - lexically expanded syntactic matching and the semantic matching with $SD_C = 1, 2, 3$ respectively. The last three strategies comprise all the features (i.e., lexical expansion, syntactic matching and semantic matching) of our concept discovery approach. They only differ in the value of the SD_C (semantic distance constraint) parameter of the concept exploration algorithm (Section III-B). The value of the path length constraint is fixed at $PL_c = 3$ for these evaluation tests.

As a result of the transformation we obtained five semantic document collections, each of which having the corresponding concept index. Table II shows for each of the concept discovery strategy: 1) the distinct number of concepts from the annotation ontology that have been involved, 2) the total number of syntactic and semantic matches, that is, the number of document units in which the concepts have been discovered by the syntactic and the semantic matching respectively and 3) the average weights of the syntactic and semantic matches calculated based on 20 randomly chosen document units. Comparing results of S_1 and S_2 which both implement only syntactic matching, we can see that the lexical expansion of concept descriptions increases the number of discovered concepts from 211 to 343 and the total number of syntactic matches from 1524 to 3182 but also the average weight of syntactic matches from 2.56 to 3.62. In other words, these increases show that the lexical expansion improves both the quantity and quality of the annotation. The next three strategies $S_3 - S_5$ produce the same number of syntactic matches as S_2 (i.e., 3182), since the syntactic matching stays intact, but they increase the number of the semantic matches (i.e., 6714; 11102; 23716). On contrary, the average weight of the added semantic matches decreases (i.e., 2.43; 1.12; 0.27). This shows that with higher values of the semantic distance constraint (SD_c) we can get more, but less relevant semantic matches.

To evaluate the performance of the proposed semantic search we formed five queries related to the data of the evaluation document collection and asked three KEY-to-

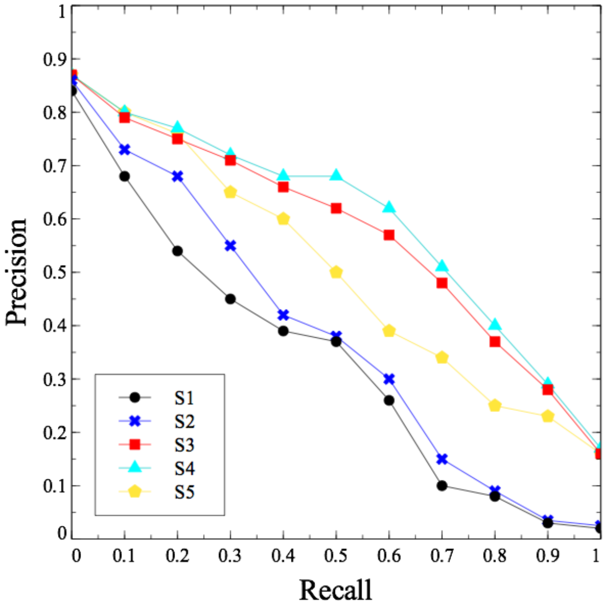


Figure 2. Interpolated precision of S_1-S_5 at standard recall points

METALS engineers to assess the relevance of document units (i.e., paragraphs) of the collection to the queries. The queries were then executed against each of the five semantic document collections. Fig. 2 shows interpolated precision at standard recall points. Comparing the P-R curves of S_1 and S_2 we can see that the lexically expanded syntactic matching outperforms from the simple syntactic matching in both recall and precision. Moreover, all three strategies (i.e, S_3, S_4, S_5) which include the semantic matching, further increase overall precision and recall. Comparing their P-R curves and by knowing that they differ only in the value of the semantic distance constraint (i.e., $SD_c = 1$, $SD_c = 2$, $SD_c = 3$) we can observe that there is an optimal value for the concept semantic distance ($SDist^c$) with regard to optimal precision and recall. It means that the semantic matches, which concept semantic distance is higher than the optimal one, reduce retrieval performances. In our evaluation the optimal value of the concept semantic distance falls in a range between 2 and 3, since the precision of S_4 is higher than of S_3 but then it drops for S_5.

The results of the preliminary evaluation indicate that the proposed concept discovery approach has potential to enlarge the amount of semantic annotations and to improve the performances of DU_s search and retrieval, not just in terms of better recall, but also in terms of better precision.

VII. Conclusions

In this paper we present an ontology-driven approach to semantic search and navigation in semantically integrated document collections. The semantic integration of document collections is achieved by the novel semantic document representation that comprises the publishing document data in RDF, the semantic annotation and indexing of document data with weighted ontological annotations and the semantic linking of related data. The results of both, the usability study and the experimental evaluation of the semantic annotation and search are promising. In the future work, we plan to continue with the evaluation of our approach, addressing issues such as the scalability, efficiency and applicability of the approach to document collections of different domains.

References

[1] J. F. Song, W. M. Zhang, W. Xiao, G. hui Li, and Z. ning Xu, “Ontology-Based Information Retrieval Model for the Semantic Web,” in *Proc. of the Int. Conf. on e-Technology, e-Commerce and e-Service*, 2005, pp. 152–155.

[2] C. Rocha, D. Schwabe, and M. P. de Aragão, “A hybrid approach for searching in the semantic web,” in *Proc. of the 13th international conference on World Wide Web*, 2004, pp. 374–383.

[3] A. Kiryakov, B. Popov, D. Manov, and D. Ognyanoff, “Semantic annotation, indexing, and retrieval,” *J. Web Sem.*, vol. 2, no. 1, pp. 49–79, 2004.

[4] D. Vallet, M. Fernández, and P. Castells, “An Ontology-Based Information Retrieval Model,” in *Proc. of the ESWC*, 2005, pp. 455–470.

[5] V. Uren and et al., “Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art,” *J. Web Sem.*, vol. 4, no. 1, pp. 14–28, 2006.

[6] J. Tuominen, M. Frosterus, and E. Hyvönen, “ONKI SKOS Server for Publishing and Utilizing SKOS Vocabularies and Ontologies as Services,” in *Proc. of the European Semantic Web Conference*, 2009, pp. 768–780.

[7] S. Nešić, “Semantic Document Model to Enhance Data and Knowledge Interoperability,” *Annals of Information Systems, Springer*, vol. 6, pp. 135–162, 2009.

[8] C. Bizer, T. Heath, and T. Berners-Lee, “Linked Data - The story so far,” *Int. Journal on Semantic Web and Information Systems*, vol. 5, no. 3, pp. 1–22, 2009.

[9] WordNet, “http://wordnet.princeton.edu/.”

[10] S. Nešić, M. Jazayeri, and D. Gašević, “Semantic Document Architecture for Desktop Data Integration and Management,” in *Proc. of the 22nd International Conference on Software Engineering and Knowledge Engineering*, 2010, pp. 73–78.

[11] Semantic Documents, “http://www.semanticdoc.org/.”

[12] S. Nešić, M. Jazayeri, M. Landoni, and D. Gašević, “Using semantic documents and social networking in authoring of course material: An empirical study,” in *Proc. of the 10th IEEE International Conference on Advanced Learning Technologies*, 2010, pp. 666–670.

[13] KeyToMetals, “http://www.keytometals.com/.”

[14] Z. Gong, C. W. Cheang, and L. H. U, “Multi-term Web Query Expansion Using WordNet,” in *Proc. of the 17th Database and Expert Systems Applications Conference, DEXA*, 2006, pp. 379–388.

Detecting Hidden Relations in Geographic Data

Ngoc-Thanh Le
Faculty of Information Technology
University of Science
Ho Chi Minh City, Vietnam
lnthanh@fit.hcmus.edu.vn

Ryutaro Ichise
Principles of Informatics Research Division
National Institute of Informatics
Tokyo, Japan
ichise@nii.ac.jp

Hoai-Bac Le
Faculty of Information Technology
University of Science
Ho Chi Minh City, Vietnam
lhbac@fit.hcmus.edu.vn

***Abstract*—The amount of linked data is growing rapidly, and so finding suitable entities to link together requires greater effort. For small data sets, it is easy enough to find entities in the data sources and link these together manually; however, doing so for large data sets is impractical. For large sets, a way is needed to discover entities and connect them automatically. In this paper, we present an algorithm to detect hidden *owl:sameAs* links or hidden relations in data sets. Since geographic names are often highly ambiguous, we used data sets comprising geographic names to implement and evaluate our algorithm. We experimentally compare our algorithm with a naïve algorithm that only uses a URI's name feature. We found that it is more accurate than the naïve algorithm in most cases, especially for resources in which there is little matching information about features.**

***Keywords*-Linked Data; Knowledge Discovery; Link Prediction;**

I. Introduction

Linked data refers to data published on the Web in such a way that it is machine-readable. It is linked to other external data sets and can in turn be linked to from external data sets [1]. Linked data uses the Resource Description Framework (RDF) to make typed statements that link arbitrary things in the world, and things are named by Uniform Resource Identifiers (URIs) and linked together by predicates.

In this paper, we mainly focus on *owl:sameAs* links. These links indicate that two URIs refer to the same thing, implying that the subject and object must be the same resource. When users create an entity to describe a thing using their own information features, if they know of other data sources on the Web that also provide information about this thing, then they can link these sources together. In this manner, the information about the thing becomes richer.

We should recognize that a linked data structure is very similar to a graph in which URIs are nodes and links are edges. Various graph algorithms exist, and the literature on them is well developed; in fact, many approaches for analyzing graphs have been extended to linked data structures [2], [3], [4]. On the basis of these observations, we decided to turn linked data into a graph upon which we can use graph mining techniques to solve the following problems.

As of 19 January 2010, the Linked Data Community estimates that the number of triples on Linking Open Data [5] is about 13 billion and the number of links is about 143 million. The amount of linked data has been growing steadily. Therefore, it may soon be difficult to find suitable entities to connect with *owl:sameAs* links. In some cases, mistakes may be made, such as linking entities that refer to different things. This means that *owl:sameAs* may be inappropriately used. In addition, a single data source may have redundant descriptions, creating confusion as to which items should be linked. Moreover, even if one manages to make an appropriate choice in some way, there is no guarantee that others will make the same choice. Finally, incorrect data affect new data in many ways. The overall effect of these problems is that information on the Web will become more and more ambiguous.

Certain data are often ambiguous; in particular, geographic names, e.g., the name of rivers, mountains, and place names of population concentrations, tend to be very ambiguous. For example, the name "Isosaai" refers to 491 places in Finland [6]. Also, there are 1724 different coordinates sharing the name "San Jose" [7] in the GeoNet and GNIS geographic name databases. Raphael Volz et al. list three types of ambiguity [7]:

1) Different geographic locations share the same name
2) One location has different names
3) A location name also stands for some other word

In our work, we are interested in geographic information and its problems. Our data set has over 2.5 million geographic names. If the above problems affect it, this would be very difficult for us to detect or resolve.

For small data sets, it is easy enough to find entities referring to the same thing in data sources and link them together manually; however, doing so for large data sets is impractical. For large sets, a way is needed to discover entities and connect them with *owl:sameAs* links automatically. The task of discovering entities can be viewed as detecting hidden relations in linked data. In other words, hidden relations are possible links that have not yet been created. The main idea behind our solution is to extract useful features by applying supervised learning on frequent graphs. We then use these extracted features to discover entities in data sources.

In brief, the contribution of this study is developing an algorithm to detect hidden relations in geographic data. The remainder of the paper is organized as follows. Section 2 briefly describes related work. The problem of detecting hidden relations and related concepts are introduced in Section 3. Section 4 describes our approach to detect hidden relations. Section 5 presents our evaluation corpus and comparatively discusses our approach's performance. We present conclusions and directions for further work in Section 6.

II. Related Work

LinkedMDB [8] demonstrates a novel way of link discovery and publishing linkage metadata to facilitate high volume and dense interlinking of RDF data sets. Because the data sources in LinkedMDB are about movies, it chooses movie titles as the feature to discover *owl:sameAs* links. Furthermore, users of LinkedMDB can give feedback on the quality of links. Because its stored attributes are information about titles and feedbacks, LinkedMDB can achieve high accuracy. However, it is not easy to apply the ideas behind LinkedMDB to other Web data sources that often mix terms of different attributes.

Silk [9] discovers *owl:sameAs* links that are used by DBpedia and by GeoNames to identify cities. Silk uses a declarative language for specifying which types of RDF links between data sources should be discovered as well as which conditions entities must fulfill in order to be linked. Depending on which data sources are linked, Silk has different thresholds ("accept" and "verify") for identifying similarity heuristics and qualifying the amounts of discovered links. This approach, however, only focuses on links of pairs of data sources: there is no guarantee that the information extracted from two data sources will enough to find suitable entities in remain data sources. In contrast to this approach, the solution we are advocating allows us to gather more information (by using data as keywords) in order to discover links.

III. Problem of Hidden Relations

What happens if data is published on the Web without *owl:sameAs* links? In such cases, each thing exists as a unique entity in a specific domain in which no two entities mention the same thing. This prevents people from contributing their own views and opinions about a thing. For example, someone talking about Mt. Fuji might describe its geographic location and climate at its peak whereas someone else might describe it as a scenic attraction. If entities such as these were not connected by an *owl:sameAs* link, a search might not return results on both of them. As a result, when users add more information about this thing, data might be duplicated. On the other hand, connecting these two descriptions by using an *owl:sameAs* link would help users to track down different information about the same resource. This means that the more *owl:sameAs* links there are, the richer the information will be.

Let us consider another scenario. When users create a new entity and want to link it to other entities with an *owl:sameAs* link, they have to find entities referring to the same resource from a mass of linked data. We call this task hidden entity detection or hidden relation detection, where the relations are *owl:sameAs* links. Hidden relations are possible links which have not yet been created. A possible link between b_y and c_y of the instance graph y in Figure 1 is an example of a hidden relation whereby c_y is found in data set C such that can be appropriately linked to b_y with *owl:sameAs*. Because there is a huge amount of linked data on the Web and it is steadily growing, it is not simple to detect such relations manually even if the entity's *domain*[1] is known.

Hidden entities can be linked to others, so we would have more sufficiently linked data after connecting these entities together. The problem is that an entity does not always link to all other entities in each domain, and the task of finding links among all domains would be extremely time consuming. Moreover, the URI identity often depends on the context in which it is used [10]; this means it is important to think about trustworthiness when creating relations among resources. That is, we need to check information describing resources in order to determine whether they are things we want to link together.

IV. Detecting Hidden Relations

A. Frequent Linked Data Graph

Linked data entities are either URIs or literals, and these are connected together by links. We can model such data as a graph. Many graph-related algorithms have been developed, and they have proven advantageous for solving a variety of problems in chemical informatics, computer vision, video indexing, and text retrieval [11]. We can consider URIs as the nodes of a graph and that all of them refer to the same resource through an *owl:sameAs* link. Because each URI is used only once per graph, URIs are represented abstractly by their *domain* name. For example, www.geonames.org is an abstraction of the URI www.geonames.org/964596. As a result, URIs having the same domain form a data set. Another reason for using *domain* name to represent URIs abstractly is that a resource in linked data often describes a type of information. The number of fields and their meaning for describing entities are treated similarly. Links among URIs are also represented as abstract entities. Abstract URIs and their links are made into an abstract graph.

Furthermore, each node represents a unique entity, and an edge describes a relationship between entities. For example, GeoNames store many name-feature relations as relational graphs. Particularly interesting among relational graphs are patterns that appear with high frequency [12] called frequent

[1] URIs have the same domain name

Algorithm 1 DHR_cSpan($g, D, local\_sups, S$)

Input: An abstract graph g, an instant graph dataset D, a set of support thresholds between any two domains *local_sups*.
Output: The closed frequent graph set S.

```
1) if $\exists g' \in S, g \subset g'$ and $support(g) = support(g')$ then
2)     return;
3) extend $g$ to $g'$ as much as possible s.t. $support(g) = support(g')$;
4) if $\exists g'$ then add $g'$ to $S$;
5) scan D, find every edge $e$ such that:
6)     $support(g' \cup \{e\}) \geq$
   minimum{local_sups of domains in graph $g' \cup \{e\}$}.
7) for each satisfied $g' \cup \{e\}$ do
8)     DHR_cSpan($g' \cup \{e\}, D, local\_sups, S$);
9) return;
```

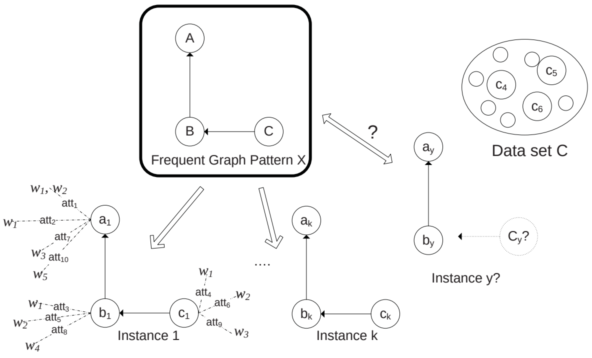


Figure 1. A hidden entity of an instance graph

patterns or graphs. Frequent graphs tend to have common relations among entities. We can extract features from the entities of such graphs and use them to identify hidden entities.

In linked data, however, the number of relation graphs is large and links are diverse. Often, there are too many frequent graphs. Because of this, it is better to mine only closed frequent graphs [12]. A frequent graph is closed if and only if there does not exist an extended graph that has the same support. The field of closed frequent graph mining has developed many algorithms, including cSpan [12], A-Close [13], CLOSET [14], CloSpan [15], and CHARM [16]. For our research, we chose to use cSpan [12] for its simplicity and efficiency in finding frequent graphs in real data. The cSpan algorithm requires choosing a support threshold for the frequency. However, we faced a problem in choosing a fixed threshold for data sets having different numbers of links. When huge data sets are connected to small data sets, it can lead to the following situation: With a fix threshold, graphs created from huge data sets tend to be very frequent because there are likely to be many links among the data. Graphs created from small data sets become relatively infrequent in comparison and hence may get dropped. For that reason, we had to modify cSpan slightly so that it could support variable thresholds. This means that, depending on which data sets are to be connected, the threshold is determined by the percentage of links between the two smallest data sets. Setting the threshold in this way enabled us to mine frequent graphs better. From here on, we shall use frequent graphs as a framework to solve our problems. Algorithm 1 (DHR_cSpan) specifies the process by which the frequent graphs are extracted. Line 6 shows the modification from algorithm cSpan of including variable thresholds.

Figure 1 illustrates a frequent graph X that has been extracted from a geographic data set on the basis of *owl:sameAs* links. There are many instances of this frequent graph $(1, \ldots, k)$. In each instance the frequent graph represents specific things. For example, supposing that the frequent graph X' includes two entities, such as Census and GeoNames, and Census links to GeoNames with an *owl:sameAs* link. Then a link from http://www.rdfabout.com/rdf/usgov/geo/us/sd/counties/perkins_county to http://sws.geonames.org/5763584/ is an instance of the frequent graph X'. Another instance is the link from http://www.rdfabout.com/rdf/usgov/geo/us/ma/counties/middlesex_county/framingham to http://sws.geonames.org/4937230/. Besides the instances of complete frequent graphs, there are graphs that lack one or more entities, such as instance y in the figure. Instance y is missing a node c_y from data set C. The reason is that c_y does not exist in this data set or there is no link to it. The way to find such missing entities is a problem that we address.

B. Attributes of the Entity

In the process of forming linked data, an RDF triple, consisting of a subject, predicate, and object, is used to represent information about resources. The subject is the URI of the described resource. The object is a literal value describing the properties of the resource or the URI of other resources. The predicate refers to links between the subject and object. Because relations in our frequent linked data graph are *owl:sameAs* links, we will consider all links except *owl:sameAs* to be attributes of the entity and the objects that are linked to as attributes' content. For example, in Figure 2, links such as *name* (link to literal value), *alternateName* (link to literal value), *inCountry* (link to URI) and even its URI name are attributes of the entity http://sws.geonames.org/283862/, whereas the *owl:sameAs* link connecting to http://dbpedia.org/resource/Gilo is not an attribute of the entity.

For the frequent graph X in Figure 1, there are three sets of attributes corresponding to three abstracted entities. The attributes' content not only describes the entity but also provides some information about the surrounding entities. Accordingly, using attributes and their content to find hidden entities is feasible. We can use useful data from

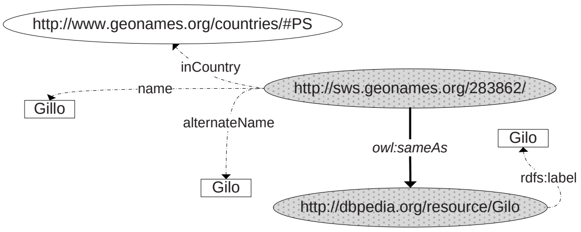

http://www.geonames.org/countries/#PS
inCountry
name
Gillo
http://sws.geonames.org/283862/
alternateName
Gilo
owl:sameAs
Gilo
rdfs:label
http://dbpedia.org/resource/Gilo

Figure 2. Attributes of an instance entity

Table I
WORD IN ATTRIBUTES OF AN INSTANCE GRAPHS

Instance Graph	Words	Store Attribute (in domain)
1	w_1	$att_1(A)$ $att_2(A)$ $att_3(B)$ $att_4(C)$
	w_2	$att_1(A)$ $att_5(B)$ $att_6(C)$
	w_3	$att_7(A)$ $att_9(C)$
	w_4	$att_8(B)$
	w_5	$att_{10}(A)$

the attributes' content as keywords for discovering entities that can be linked to it. However, the attributes of each entity in different domains vary in quantity and quality; even entities in the same domain will have such differences. Moreover, not all attributes are useful for finding hidden entities. Therefore, choosing only the most useful attributes is a prerequisite for creating a hidden relations detection algorithm.

C. Choosing Useful Attributes

The data set has information related to geographic names. As a result, we chose the feature "word" (lexical) for identifying useful attributes. The feature "word" in our paper is a sequence of characters separated by spaces. Our assumption is that entities are linked when the contents of their respective attributes have at least one word in common. This means that they mention the same concept. In Figure 2, for example, the entities http://sws.geonames.org/283862/ and http://dbpedia.org/resource/Gilo have common word "Gilo" in the attributes *alternateName*, *URI name* of the DBpedia entity and *rdfs:label*. Hence, word "Gilo" seems to be useful information for identifying the described resource. By collecting such words, we should be able to find related entities more easily. The question is, into which attributes are these words often distributed? If this question can be answered, it means that we have useful attributes. To achieve this, we should collect the words and the attributes containing those words in each instance graph. Words that do not appear in all of the entities of a graph will be removed from further consideration.

Table I shows the words extracted from attributes of the first instance graph in Figure 1, where att_i for i = $\{1, 2, \ldots\}$ are attributes of the entities of the graph, and w_j for j = $\{1, 2, \ldots\}$ are words extracted from the attributes. Since each entity belongs to a specific domain, we consider its attributes to be domain attributes. Words w_3, w_4, and w_5 do not appear in all entities of the graph. Therefore, they are removed from further consideration. Other instance graphs are similarly processed. The result is a large table of words and attributes. Our goal is to seek feature attributes that can be used to extract content for predicting hidden relations. Accordingly, we rank attributes by increasing their weight one unit whenever they appear on the word table of the instance graphs. For example, in Table I, the first attribute appear two times, so its rank is 2. If the first attribute appears three times in the second instance graph, its rank becomes 5, and so on. Attributes that exist in many instance graphs will certainly have higher ranks than ones that only exist in a few instance graphs. Such high ranking attributes play a major role in detecting hidden relations. However, we need to consider that some attributes might be useful in some graphs but useless in other graphs. In some cases, attributes can even cause noise. Therefore, we use a threshold to reduce the number of bad attributes. Attribute rank can not go lower than the threshold. Such threshold is selected to maximize the accuracy of our approach.

Since the number of instances in each frequent graph is not the same, rank values might be quite different for different frequent graphs. In order to compare the correlation of attributes among frequent graphs as well as reduce calculating cost in later calculations, we use attribute weight instead of rank. Attribute weight is calculated from rank as follows:

$$weight_i = \frac{rank_i}{N(X)}, \tag{1}$$

where $rank_i$ is the rank of the ith attribute, and $N(X)$ is the number of instance graphs of frequent graph X. In the next section, we use the above feature attributes and their weights for finding hidden relations.

D. Distance Estimation

Here, a graph lacking an entity is a graph that is missing one entity compared with some frequent graph. A graph missing more than one entity can be dealt with recursively. That is, after we find the first entity, we look for the second entity, and so on. In Figure 1, the instance graph y consists of two entities a_y and b_y, and a missing entity c_y. Our task is to find c_y in data set C, where C is the set of entities having the same domain as c_y. In fact, entity c_y may not exist in C.

Words in the feature attributes of entities are extracted. Entities such as a_y and b_y existed in the instance graph

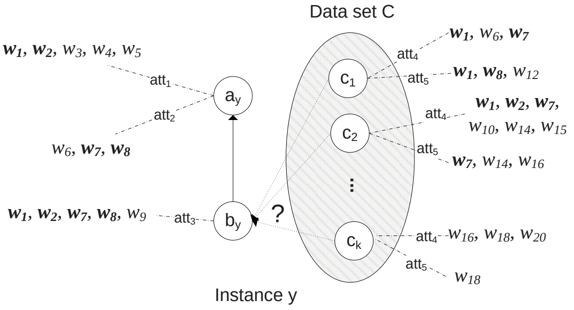


Figure 3. Words appearing in feature attributes of entities

y in Figure 1, and so the words in these entities are fewer and easier to extract. However, c_y has not yet been identified, and so we have to list all the words in each feature attribute belonging to data set $C(c_1, c_2, \ldots, c_k)$. This process consumes much time and computing resources. To reduce the burden, we index words in C and only extract words that appear in both a_y and b_y. After extracting words from the feature attributes, we remove words that do not exist in all entities. This means that we only keep words that appear in all entities of the instance graph. Note that we propose another solution for the case in which no such word exists (see the end of this section). Figure 3 illustrates words extracted from the feature attributes, $\{att_1$ (domain A), att_2 (domain A), att_3 (domain B), att_4 (domain C), att_5 (domain C)$\}$, in the instance graph y and entities in data set C.

Entity a_y and entity b_y share the word set $\{w_1$, w_2, w_7, $w_8\}$. Thus, entity c_y that will be detected must store the word subset $\{w_1$, w_2, w_7, $w_8\}$ in the content of its feature attributes. Let S_t be the set of words appearing in all entities of the graph after entity t has been inserted. In Figure 3, we have $S_1 = \{w_1, w_7, w_8\}$; $S_2 = \{w_1, w_2, w_7\}$; $\ldots$; $S_k = \{\emptyset\}$. The distance is estimated using the attribute weights and the number of words stored in S_t after S_t is projected in turn onto these attributes. For example, the set S_1 after being projected onto att_4 becomes the set $\{w_1, w_7\}$, and so the number of words in the projected S_1 is 2. For each entity c_t in data set C, the distance from it to the graph is defined as

$$l(c_t) = \frac{1}{\sum\limits_{i=1}^{n}[weight_i \times N(\pi_{att_i}(S_t))]}, \qquad (2)$$

where *n* is the number of feature attributes in the discovered domain, $weight_i$ is the weight of the *i*th feature attribute, $\pi_{att_i}(S_t)$ is the projection of the set S_t onto attribute att_i, and $N()$ is a function to count the number of words in the projected S_t. Because S_t contains words extracted from many different attributes, the projection $\pi_{att_i}(S_t)$ is a way to pick out words only from attribute att_i. Accordingly, the shorter the distance l of the entity is, the more suitable the entity will be to link to the graph. However, there are likely entities in C that have equal shortest distances. Therefore, we have to decide which among them should be linked to the graph next. Figure 4 shows an example of this problem. The entities c_1 and c_2 have equal distances l (i.e., $l(c_1) = l(c_2)$). Thus, we need to determine which, c_1 or c_2, is more suitable for connecting to the graph.

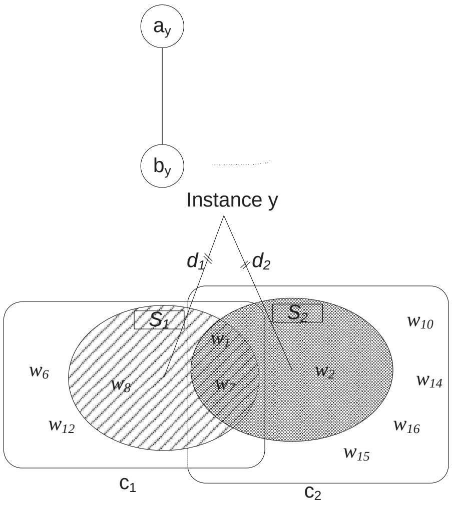


Figure 4. Two entities having the same distance to a graph

Words appearing in a set of feature attributes are not involved in the calculation if they do not occur in all entities of the graph. In Figure 4, these words are $w_6, w_{10}, w_{12}, w_{14}, w_{15}$, and w_{16}. These words can cause entities to become irrelevant. This means that the entity containing more words not in S_t will have a larger distance. The above considerations motivated us to use the following function:

$$d(c_t) = l(c_t) - \varepsilon\frac{1}{N(\overline{S_t})}, \qquad (3)$$

where ε is a small positive number such that this measure does not affect the main distance $l(c_t)$. The resulting set from using distance d does not add any entities beyond those added using distance l. $\overline{S_t}$ is the complement of S_t (i.e., words in the content of the feature attributes of c_t do not appear in the whole graph). From this definition, we can see that the shorter the distance d of the entity is, the more suitable the entity will be for linking to the graph. Note that after this procedure, if there are still many entities with the same shortest distance d, then we must choose among them randomly or manually. Our experiment showed that this approach improved accuracy in comparison with simply using the distance l in Equation 2.

Next, we resolve the problem of entities in the graph that do not share words in different domains because of irrelevant feature attributes. In this case, the distance d from each entity

Algorithm 2 DHR-DE($g, C, P, pWeight, g'$)

```
Input: a graph $g$, a dataset $C$ for finding entity, and
a set of feature attributes $P$ and their weight $pWeight$.
Output: a graph $g'$ which added found entity in $C$ to $g$.
 1) Set current shortest distance $min\_d = -1$;
 2) for each $c_t \in C$ do
 3)    Extend $g$ to $g'$ by adding $c_t$ in graph $g$;
 4)    $S_1$ = extract words stored in feature attributes of $g'$
       that appear in all entity of $g'$;
 5)    $S_2$ = extract words stored in feature attributes of $g'$
       that do not appear in all entity of $g'$;
 6)    if $S_1 \neq \{\emptyset\}$ then
 7)       Calculate distance $d$ using $S_1$ and $S_2$.
 8)       if $min\_d = -1$ or $min\_d > d$ then
 9)          $r = \{\emptyset\}$; $min\_d = d$;
10)          Insert $c_t$ into $r$;
11)       if $min\_d = d$ then
12)          Insert $c_t$ into $r$;
13)    else
14)       if (graph $g$ contains one node) then return;
15)       else
16)          Split graph $g$ into subgraphs and execute
             lines 4 to 7 for each distance from a subgraph;
17)          Calculate distance $d'$ from each subgraph distance;
18)          Insert $c_t$ into $r$ if distance $d'$ is less than or
             equal to $min\_d$;
19) if ($r$ contains more than one entity) then
20)    Choose and entity randomly;
21) Extend $g$ to $g'$ by adding such entity in graph $g$;
22) return;
```

in data set C to the graph is zero. One idea is to consider each entity in a graph as a separate subgraph and the distance from an entity in data set C to the graph equals the sum of distances from it to the subgraphs:

$$d'(c_t) = \sum_{j=1}^{m} d_j(c_t), \qquad (4)$$

where m is the number of entities in the graph, and d_j is the distance from c_t to a subgraph that stores only one *j*th entity. Suppose that entity a_y and entity b_y in Figure 3 do not share any word, and so S is always empty. To estimate the distance, we view instance graph y from a different angle: y includes two subgraphs, one storing entity a_y and one storing entity b_y. Consequently, the distance from c_t to the graph is the sum of distances d from c_t to the subgraph storing a_y and from c_t to the subgraph storing b_y. Algorithm 2 illustrates the process used to find the most suitable entity in dataset C using distance functions from Equation 2 and 4.

V. Evaluation

We evaluated the proposed algorithm on real data sets. The data sets were derived from four publicly available geographic information sources:

The U.S. Census data is provided by the Census Bureau. The Census data comprises population statistics at various geographic levels, from the United States as a whole, to state, county, sub-county (roughly, cities and incorporated towns), so-called “census data places”, ZIP Code Tabulation Areas (ZCTAs, which approximate ZIP codes), and even deeper levels of granularity. The data set contains around 3,200 counties, 36,000 towns, 16,000 villages, and 33,000 ZCTAs [17].

GeoNames gathers geographical data, such as names of places in various languages, elevations, and populations, from various sources. All lat/long coordinates are in WGS84 (World Geodetic System 1984). It contains over 8 million geographical names and consists of 7 million unique features including 2.6 million populated places and 2.8 million alternate names [18].

The DBpedia data set is a large multi-domain ontology which has been derived from Wikipedia. The DBpedia data set contains geo-coordinates for 392,000 geographic locations [19].

The World Factbook provides information about the history, people, government, economy, geography, communications, transportation, military, and transnational conflicts of 266 world entities [20].

The above data sources were linked together with *owl:sameAs* links, creating about 100,000 connected graphs. Note that not every entity had *owl:sameAs* links; these formed empty graphs, and we did not include them in our graph set. We applied our modified cSpan to find frequent graphs in the graph set. A 20% link threshold between datasets was used. With these settings, we derived 13 frequent graphs patterns. These frequent graphs were used in the following evaluations.

To test the quality and validity of our distance measure based on feature attributes, we compared our algorithm for detecting hidden relations with a naïve algorithm. The naïve algorithm used only information about the URI name to make a prediction. We used a *k-fold* cross-validation method with $k = 10$ [21] to construct the training and test sets. That is, the dataset was split into 10 equal groups. In turn, each group was used for testing and the remaining groups were used for training. The final result is an average over choices. For each instance of a frequent graph , we evaluated the accuracy by removing one entity and attempting to find it again. Note that not all frequent graphs included all four domains (i.e., US Census, GeoNames, DBpedia, and World Factbook), so the choice of entity to be removed depended on whether it existed in the graph. Also note that the frequent graphs were directed graphs and did not have any ambiguities. Figure 5 lists the frequent graphs with the number of instances.

Figure 6 compares the accuracies of our algorithm and the naïve algorithm. Accuracy is the precision of prediction, i.e., the percentage of found entities that were correct. In the case in which we removed an entity belonging to the US Census, GeoNames, or DBpedia domain, our method

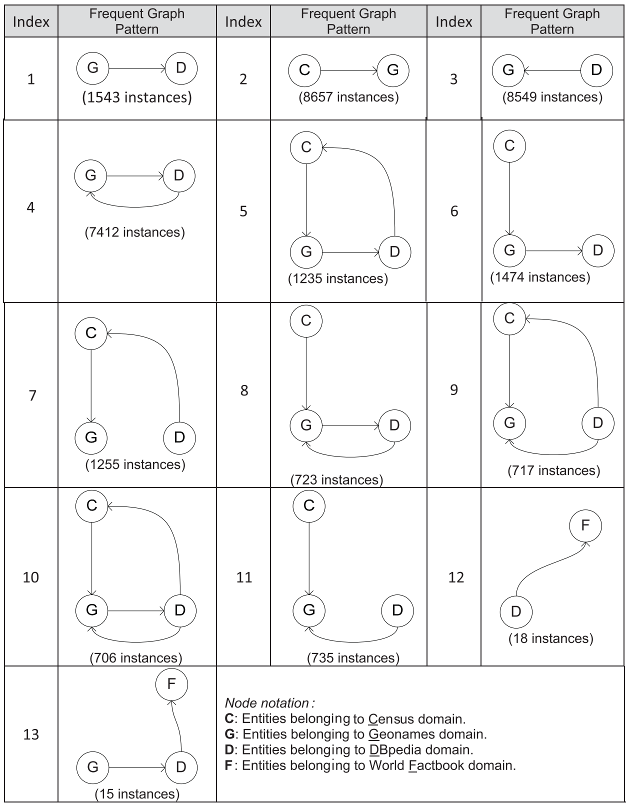


Figure 5. Closed frequent graphs

gave a better result than the naïve method. For the World Factbook, however, our method gave worse results because the names in the URIs were too well matched. For example, http://dbpedia.org/resource/Nauru and http://www4.wiwiss.fu-berlin.de/factbook/resource/Nauru match "Nauru". In our algorithm, information extracted from other feature attributes caused significant noise. However, we are only interested in the general case wherein the attributes of entities do not yield very similar information. In addition, the results for the first and third frequent graphs patterns were quite low. The reason is that if one of the entities is missing, then the information gained from the feature attributes of the other entities is not enough to detect the missing one.

Finally, we considered graphs of data sets that really were missing entities in the data and tried to predict new entities that could be linked to them. Our algorithm was able to find new entities even though the number of such graphs was very small. This task was difficult because the entities may not exist in the data set. For example, in the case of http://sws.geonames.org/5879092/ and http://www.rdfabout.com/rdf/usgov/geo/us/ak in the linked data, the two entities are linked by *owl:sameAs* and do not link to any other entity. They both refer to Alaska. Our method found a new entity in DBpedia which can link to them: http://dbpedia.org/resource/Alaska

Looking at the results, we can see that our method generally increased the completeness of the linked data. Although it was far from perfect, it easily incorporated new knowledge with few mistakes.

VI. Conclusion

We presented an approach to detecting hidden *owl:sameAs* relations in geographical data sets, such as those of the U.S. Census, GeoNames, DBpedia, and World Factbook. Since feature attributes play an important role in describing a resource, we can carry over relationships between resources. Our approach uses supervised learning to train a feature attribute set and uses the set for detecting relations. We compared the outcomes of ours and a naïve approach using only URI name data for discovering hidden relations and found that our approach has higher accuracy in most cases, especially for resources in which there are not too many matching feature attributes.

There are still many interesting aspectss to be studied in detecting relations. One of them is noise. Besides useful information, there is also superfluous information, or noise. Such noise does not describe resources, and so it makes the distance estimation worse. For example, articles, prepositions, and auxiliary verbs occur frequently, but they do not help in detecting hidden relations.

VII. Acknowledgment

We would like to thank members in NII for discussing our research and giving us valuable recommendations. We also thank the reviewers for their comments.

References

[1] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data - the story so far," *International Journal on Semantic Web and Information Systems*, vol. 5, pp. 1–22, 2009.

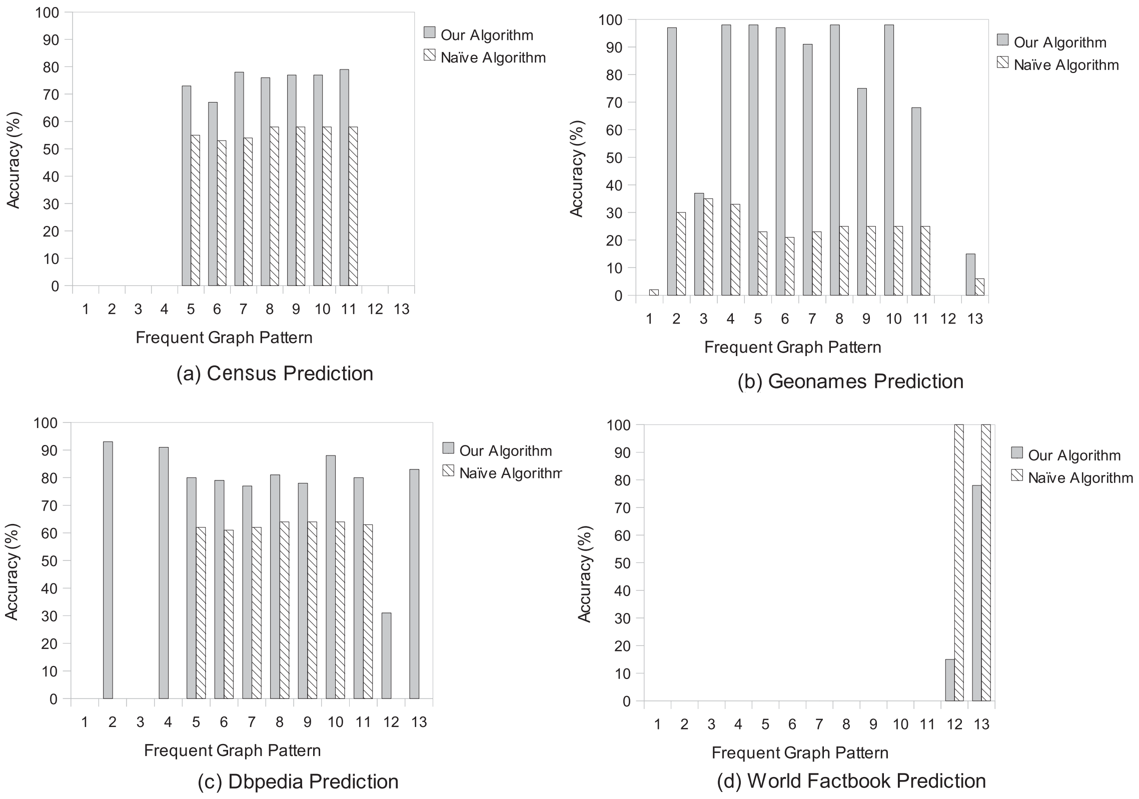
[2] J. Mi, H. Chen, B. Lu, T. Yu, and G. Pan, "Deriving similarity graphs from open linked data on semantic web," in *Proceedings of the 10th IEEE International Conference on Information Reuse and Integration*, 2009, pp. 157–162.

[3] P. Cudre-Mauroux, P. Haghani, M. Jost, and K. A. H. de Meer, "idMesh: Graph-based disambiguation of linked data," in *Proceedings of the 18th International World Wide Web Conference*, 2009, pp. 591–600.

[4] L. Getoor, "Link mining: A new data mining challenge," *ACM SIGKDD Explorations Newsletter*, vol. 5, pp. 84–89, 2003.

[5] Linked Data Community, "Statistics on data sets of LOD," http://esw.w3.org/topic/TaskForces/CommunityProjects/LinkingOpenData/DataSets/Statistics.

[6] E. Hyvonen, R. Lindroos, T. Kauppinen, and R. Henriksson, "An ontology service for geographical content," in *Proceedings of the 6th International and 2nd Asian Semantic Web Conference*, 2007, pp. 33–34.



(a) Census Prediction

(b) Geonames Prediction

(c) Dbpedia Prediction

(d) World Factbook Prediction

Figure 6. Accuracies of our algorithm and naïve algorithm

[7] R. Volz, J. Kleb, and W. Mueller, “Towards ontology-based disambiguation of geographical identifiers,” in *Proceedings of the WWW2007 Workshop i3: Identity, Identifiers, Identification*, 2007.

[8] O. Hassanzadeh and M. Consens, “Linked movie data base,” in *Proceedings of the WWW2009 Workshop on Linked Data on the Web*, 2009.

[9] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov, “Silk - a link discovery framework for the web of data,” in *Proceedings of WWW2009 Workshop on Linked Data on the Web*, 2009.

[10] A. Jaffri, H. Glaser, and I. Millard, “URI disambiguation in the context of linked data,” in *Proceedings of the WWW2008 Workshop on Linked Data on the Web*, 2008.

[11] D. J. Cook and L. B. Holder, *Mining Graph Data*. John Wiley and Sons, 2007.

[12] X. Yan, X. J. Zhou, and J. Han, “Mining closed relational graphs with connectivity constraints,” in *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005, pp. 324–333.

[13] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, “Discovering frequent closed itemsets for association rules,” in *Proceedings of the 7th International Conference on Database Theory*, 1999, pp. 398–416.

[14] J. Pei, J. Han, and R. Mao, “Closet: An efficient algorithm for mining frequent closed itemsets,” in *Proceedings of 2000 ACM-SIGMOD International Workshop Data Mining and Knowledge Discovery*, 2000, pp. 11–20.

[15] X. Yan, J. Han, and R. Afshar, “Clospan:mining closed sequential patterns in large datasets,” in *Proceedings of 3rd SIAM International Conference on Data Mining*, 2003, pp. 166–177.

[16] M. J. Zaki and C.-J. Hsiao, “Charm: An efficient algorithm for closed itemset mining,” in *Proceedings of 2nd SIAM International Conference on Data Mining*, 2002, pp. 457–473.

[17] J. Tauberer, “The U.S. census data,” http://www.rdfabout.com/demo/census/, 2007.

[18] M. Wick, “The GeoNames geographical database,” http://www.geonames.org/.

[19] DBpedia Team, “The DBpedia database,” http://wiki.dbpedia.org/, 2009.

[20] CIA Factbook D2R Server, “The World Factbook database,” http://www4.wiwiss.fu-berlin.de/factbook/.

[21] G. J. McLachlan, K.-A. Do, and C. Ambroise, *Analyzing microarray gene expression data*. John Wiley and Sons, 2005.

Context-based Hybrid Method for User Query Expansion

Ounas Asfari, Bich-Lien Doan, Yolaine Bourda
SUPELEC/Department of Computer Science
University of Paris-Sud 11
Gif Sur Yvette Cedex, France
{name.surname}@supelec.fr

Jean-Paul Sansonnet
LIMSI-CNRS
University of Paris-Sud 11
Orsay cedex France
jps@limsi.fr

***Abstract*— Today, there is a real challenge in accessing relevant information on the Web according to the user's needs and the context. There are always certain needs behind the user query and these queries are often ambiguous and shortened (especially in the case of mobile users), thus we need to handle the user queries intelligently to provide personalized results in a particular context. For improving user query processing, we present a context-based hybrid method for query expansion that automatically generates context-related terms. It considers the context as the actual state of the task that the user is undertaking when the information retrieval process takes place. The method uses the UML state diagram for modeling the current task and for detecting the transitions at time intervals with the task state changes. Furthermore, we introduce a new concept of SRQ (State Reformulated Queries), which is used to reformulate queries according to the user task context and the ontological user profile. Using experimental study, our approach has proved its relevance for certain contexts, the preliminary results are promising.**

Keywords- query reformulation; context; task modeling; Information Retrieval; user profile.

I. INTRODUCTION

The Internet offers almost unlimited access to information of all kinds. As the volume of the heterogeneous resources on the web increases and the data becomes more varied, massive response results are issued to user queries. Thus, large amounts of information are generated in which it is often difficult to distinguish relevant information from secondary information or even noise. Recent studies have tried to dynamically enhance the user query with the user's preferences by creating a user profile for providing personalized results [1]. However, a user profile may not be sufficient for a variety of queries of the user. For example a tourist and a programmer may use the same word "java" (Java Island in Indonesia, Java programming language, the Java Coffee, etc.), in some situations the programmer may need information about the Java island that is not found in his preferences. One disadvantage of automatic personalization techniques is that they are generally applied out of context. So, not all of the user interests are relevant all of the time, usually only a subset is active for a given situation, and the rest cannot be considered as relevant preferences.

On the other hand, new devices are constantly appearing and becoming a principle part of our daily lives. the multitude of devices (PC, PDA, cellular phone, etc.) including diverse platforms, the different user knowledge levels, characteristics and expectations, and the various work environments, have created new considerations and stakes to be satisfied [2]. To overcome the previous problems, studies taking into account the user context are currently undertaken. As a result, the information needs of mobile users are related to contextual factors such as user interests, user current task, location, direction, etc.

The user context can be assimilated to all factors that can describe his intentions and perceptions of his surroundings, these factors may cover various aspects: physical, social, personal, professional, technical, task, etc. Fig. 1 shows these factors and examples for each one [3].

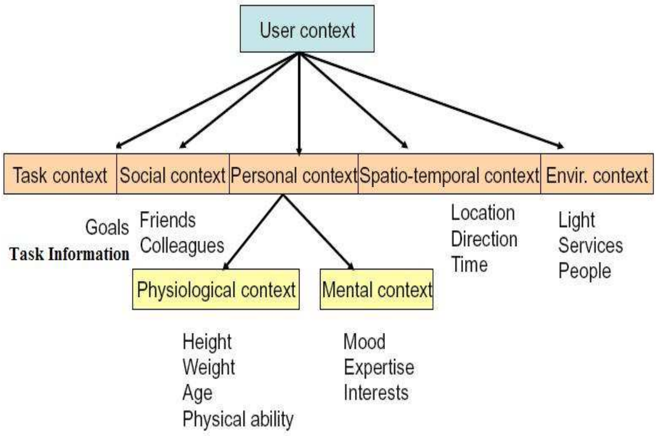


Figure 1. A context model from Kofod-petersen.

The problems to be addressed here include how to represent the context, how to determine it at runtime, and how to use it to influence the activation of user preferences. It is very difficult to take into consideration all the contextual factors in one information retrieval system, so the researchers often define the context as certain factors (location for example).

Thus in this paper our definition of the context is that the context describes the user current task, its changes over time and its states, i.e., we take into account the user current task which the user is undertaking when the information retrieval process occurs.

Queries, especially short one, do not provide a complete specification of the information need. Many relevant terms can be absent from queries and terms included may be ambiguous. Typical solution includes expanding query representation by exploiting semantic resources [4] or user profile [5]. That refers to methods of query reformulation, i.e., any kind of transformation applied to a query to facilitate a more effective retrieval.

This paper present a method to reformulate user queries depending on the user profile, containing his interests, together with the user context which is considered as the actual state of the user current task in order to provide personalized results in context. Moreover we will consider that the user queries are related to the task at hand, indeed that are part of it. We combine knowledge about query (linguistic knowledge, using WordNet and semantic knowledge using ODP ontology, Open Directory Project, www.dmoz.org) and knowledge about user (user profile and user task context) into a single framework in order to provide the most appropriate answer for a user's information needs in the search time and task state.

For example, if a user has to organize a workshop, many states for this task exist, such as the choice of the workshop topics and the choice of the program committee members, etc. Submitting two equivalent queries in two different states, the relevant results to each task state will be different, so the proposed system has to provide the different relevant results to each state.

The rest of the paper is organized as follows: Section 2 shows the related work; Section 3 introduces the models and algorithms to reformulate user's queries; section 4 presents the architecture of our system; Section 5 shows the experimental study and examples Finally, Section 6 gives the conclusion and future work to be done.

II. Related Work

Query expansion is the process of augmenting the user's query with additional terms in order to improve results by including terms that would lead to retrieving more relevant documents. Many works have been done for providing personalized results by query reformulation.

Two main approaches based on the user profile to reformulate a query have been proposed: query enrichment process which consists in integrating elements of the user profile into the user's query [6], the user profile is defined as a list of disjunctive predicates, including selections and joints. Given such a profile, the query enrichment process consists in reformulating the initial user query by adding predicates from this profile. The second approach based on a user profile is the query rewriting process which translates the query to access the real data sources [7].

The limitation of these approaches is that they do not take into consideration the user context for activation the elements from the user profile.

Studies on query reformulation by relevance feedback are proposed, the aim is to use the initial query in order to begin the search and then modify it from the judgments of the relevance and irrelevance to the user. The new complaint obtained in each iteration feedback, can rectify the direction of the research [8]. Because relevance feedback requires the user to select which documents are relevant, it is quite common to use pseudo-relevance feedback.

Furthermore the techniques of disambiguation aim to identify precisely the meaning referred by the terms of the query and focus on the documents containing the words quoted in the context defined by the corresponding meaning [9]. But this disambiguation may cause the query to move in a direction away from the user's intention. For example the query “windows” might be about actual windows in houses or the Microsoft Windows operating system. A system might choose an interpretation different from the user's intention and augment the query with terms related to the wrong interpretation.

Many approaches like [4] try to reformulate the web queries based on semantic knowledge about different application domains from Research-Cyc for example, others use sense information (WordNet in general) to expand the query [10].

Many approaches, for example [11], expand the user initial query by using ontology in order to extract the semantic domain of a word and add the related terms to the initial query. But sometimes these terms are not related to query terms. More precisely they are related to the query but only under a particular context of the specific query.

This paper presents a new approach for improving user query processing. We propose a hybrid query expansion method that automatically generates query expansion terms from the user profile and the user task. In our approach we exploit both a semantic knowledge (ODP Ontology) and a linguistic knowledge (WordNet) to learn the user's task, and we exploit an UML states diagram for one task to learn user current state.

III. Models and Algorithms

Our aim is to provide context-based personalized results. For that, we improve the user web-queries intelligently to address more of the user's intended requirements. We generate a new query language model for the purpose of query reformulation based on the user context and an ontological user profile. We consider the user current task as a contextual factor. Here we will describe our models for detecting the user current task, constructing an ontological user profile and generating the reformulated queries.

A. General Language Model

We construct here a new general language model for query expansion including the contextual factors and user profile in order to estimates the parameters in the model that is relevant to information retrieval systems. In the language modeling framework, a typical score function is defined in KL-divergence as follows [15]:

$$Score\,(Q, D) = \sum_{t \in V} P(t \mid \theta_Q) \log P(t \mid \theta_D) \propto -KL(\theta_Q \parallel \theta_D) \qquad (1)$$

Where: θ_D is a language model created for a document *D*, θ_Q a language model for the query *Q*, generally estimated by relative frequency of keywords in the query, and V the vocabulary.

$P\,(t|\theta_D)$: The probability of term t in the document model,

$P\,(t|\theta_Q)$: The probability of term t in the query model,

$P\ (Q \mid D) = \prod P\ (t \mid \theta_D)^{c\,(t\,;Q)}$ c(t ;Q) : Frequency of term t in query Q;

The basic retrieval operation is still limited to keyword matching, according to a few words in the query. To improve retrieval effectiveness, it is important to create a more complete query model that represents better the information need. In particular, all the related and presumed words should be included in the query model. In these cases, we construct the initial query model containing only the original terms, and a new model SRQ containing the added terms. We generalize this approach and integrate more models for the query. Let us use θ_Q^0 to denote the original query model, θ_Q^T for the task model, θ_Q^S for the contextual state model, and θ_Q^U for a user profile model. θ_Q^0 can be created by MLE (Maximum Likelihood Estimation)[3].

Given these models, we create the following final query model by interpolation:

$$P\,(\mathrm{t} \mid \theta_Q) = \sum_{i \in X} \alpha_i\, P\,(t \mid \theta_Q^i) \qquad (2)$$

Where: X= {0, T, S, U} is the set of all component models and a_i (with $\sum_{i\in X} a_i = 1$) are their mixture weights.

Thus the (1) becomes:

$$Score\ (Q,D) = \sum_{t\in V}\sum_{i\in X} \alpha_i P(t \mid \theta_Q^i) \log P(t \mid \theta_D) = \sum_{i\in X} \alpha_i Score_i\ (Q,D) \qquad (3)$$

where:

$$Score_i\ (Q,D) = \sum_{t\in V} \alpha_i P(t \mid \theta_Q^i) \log P(t \mid \theta_D) \qquad (4)$$

is the score according to each component model.

The remaining problem is to construct task model, contextual model and user profile model and to combine all the models.

B. Constructing Task Model

The task model is used to detect and describe the task performed by the user, when he submits his query to the information retrieval system. We consider the task as the contextual factor of the user. In this paper we depend on study questionnaires [16], which were used to elicit tasks that were expected to be of interest to subjects during the study. A generic classification was devised for all tasks identified by all subjects, producing the following nine task groupings:

Academic Research; News and Weather; Shopping and Selling; Hobbies and Personal Interests; Jobs/Career/Funding; Entertainment; Personal Communication; Teaching; Travel.

For example, the task labels “viewing news,” “read the news,” and “check the weather” would be classified in Group 2: “News and Weather.”

We generate a UML states diagram for each task in order to detect the changes in the task-needs over time and for describing all the sequences of the performed task. This generated diagram contains the task states and at least one attribute for each one. Accordingly, an index is built for: the terms of the tasks, the terms of its states including the state attributes, and the related task concepts from ODP. Thus this index consists of r terms. We will use this index when using the term vector model.

The user task can be identified in two different ways:

1) Manually, by the user who selects one task from the proposed tasks and assigns the selected task to his queries.

2) Automatically, by taking advantages of existing linguistic (WordNet) and semantic resources (ODP Ontology) for assigning a task to user query.

Here, we use the second way in order to facilitate the process to users. For applying the second way, we apply the following *algorithm*:

Let q be a query submitted by a specific user at the current task denoted A*. This query is composed of n terms; it can be represented as a single term vector:

$$\vec{q} = \langle t_1, t_2, \ldots, t_n \rangle$$

For this query $\vec{q}$ a current task A* is built by a single term vector:

$$\vec{A}_* = \langle a_{s1}, a_{s2}, \ldots, a_{si} \rangle$$

Where: a_{S1}, a_{S2}, …a_{Si} the terms that represent the state attributes of the task states s_1, s_2, …s_i for the current task A*. For example, if the actual state is “Find a Restaurant”, then the state attribute will be “Restaurant” and a value from the user profile (such as vegetarian) will be assigned to this state attribute in order to personalize the query.

The initial query q is parsed using WordNet in order to identify the synonymous terms and to build the baseline query:

$$\vec{q_w} = \langle t_{w1}, t_{w2}, \ldots, t_{wn} \rangle$$

The baseline query $\vec{q_w}$ is queried against the ODP ontology in order to extract a set of concepts (c_1,c_2…,c_m with m≥n) that reflect the semantic knowledge of the user query. These concepts of the user query and its sub-concepts are represented as a single term vector

$$\vec{C_q} = \langle c_1, c_2, \ldots, c_m \rangle$$

Then the concepts are compared with the previous nine tasks, to do this, we compute the similarity weight between $\vec{C_q}$ and the proposed nine tasks, depending on the task index which is previously explained:

$$SW\,(A_1) = \mathrm{Cos}\ (\vec{C_q}\,,\ \vec{A_1}\,)$$

$$SW\,(A_2) = \mathrm{Cos}\ (\vec{C_q}\ \ ,\vec{A_2}\,)$$

…….

....

$$SW\,(A_9) = \mathrm{Cos}\ \ (\vec{C_q}\ \ ,\vec{A_9}\,)$$

Finally, the task $\vec{A_*}$ corresponding with the maximum similarity weight (*Max* (*SW* (*A**))) is automatically selected as the current task. Fig. 2 shows the various vectors.

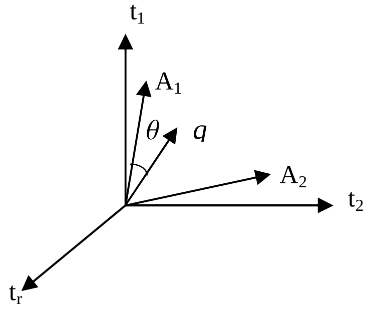


Figure 2. Representation of the tasks and the query as term vectors.

Where: query terms: $t_1, t_2, \ldots, t_n$.

Terms of task index: $t_1, t_2, \ldots, t_r$.

Terms of task state attributes: $a_{S1}, a_{S2}, \ldots, a_{Si}$.

Each term's weight is computed using tf * idf weighting scheme.

For example if the user submits the query *q= {Tourism in Toulouse}*, then the steps of our approach for detecting the user task are shown in Table 1:

TABLE I. APPLYING TASK MODEL TO THE QUERY Q

Description	Knowledge used	Result
parsing the initial query using WordNet	WordNet	A set of query terms (t_1,.., t_n) (tourism, Toulouse) and its synonymous terms (that will be used as the baseline query(services to tourists, touring, travel, city in France)
The concepts in ontology that represent the baseline query terms are identified.	Ontological information from ODP ontology.	A set of the concepts ($c_1, \ldots, c_m$ with $m \geq n$) relevant to the baseline query. (Travel Guides, Travel and Tourism, Vacations and Touring, Touring Cars, Weather, Food, Maps and Views, hotel, University of Toulouse, Commerce and economy)

So, the task that assigned to the user query *q* is: "*travel*" as it has the most similarity weight number.

C. Contextual State Model

The contextual state model is responsible for determining and analyzing the actual state of the current task. We suppose that the different states of the current task are modeled using an UML state diagram. There is at least one relevant attribute as_i for each detected state S_i. Because mobile device moves with the user, it is possible to take into account the actual task state in which the user is in when submitting certain queries to the information retrieval system IRS. Such contextual information may come automatically from various sources such as the user's schedule, sensors, entities that interact with the user; it may also be created by the user.

According to our assumption, we have defined 9 UML state diagrams for the main pre-defined tasks. After the user's query is submitted to our platform, the related task is assigned automatically to the user query and a set of SRQ (State Reformulated Queries) related to each state is presented to the user. The user is then asked to choose the appropriate SRQ according to his state. Finally, the contextual model will follow the UML state diagram to present the next SRQ.

D. Ontological user profile model

Ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts so the basic building blocks of ontology are concepts and relationships. Concepts (or classes or categories or types) appear as nodes in the ontology graph.

A *user profile* is a collection of personal data associated to a specific user. The *Ontological user profile* is constructed by the representation of the user profile as a graph of related concepts of the ODP ontology, inferred using an index of user documents. Here, a dynamic ontological user profile is considered as semi-structured data in the form of attribute-value pairs where each pair represents a profile's property.

The properties are grouped in categories or concepts using ODP Ontology, this allows us to help users to understand relationships between concepts, moreover, to avoid the use of wrong concepts inside queries. e.g., for a query "looking for a job as a Professor", ontology suggests relevant related terms: teaching, research etc. for example in the proposed ontological user profile we can find global category (language, address, age…etc.) and local category (preferences of restaurants, hotel, travel, music, videos, etc.), i.e. the annotating of each concept in ODP ontology is done by giving value for each attribute in the ontology concept based on an accumulated similarity with the index of user documents, a user profile is created consisting of all concepts with non null value.

Using ontology as the basis of the profile allows the initial user behavior to be matched with existing concepts in the domain ontology and relationships between these concepts [12]. When the ontological user profile is created, its query-related concepts must be activated. This is done by mapping the query context $C_q = \langle c_1, c_2, \ldots, c_m \rangle$ on this ontological user profile (note that, the query context is calculated during the construction of the task model). This allows to activate for each query context concept its semantically related concepts from the ontological user profile, following our contextual approach depending on the relevant propagation [13]. Hence, the relevant user profile attributes that are determined by the previous activated concepts are found. This attributes with its values are used to reformulate the user query.

E. SRQ Model (State Reformulated Queries)

Query expansion is the process of adding relevant terms to the original query [14]. However, in a more general sense, it also refers to methods of query reformulation, Thus we look for a relevant terms to use it in query expansion. But what do we mean by *relevant terms*?

The terms are relevant if they are related to the query, the user, and the task state in the same time and don't contain unrelated terms. The initial user query is reformulated depending on these relevant terms in order to produce SRQ (State Reformulated Query) to improve the retrieval performance. The two aspects for producing SRQ are query expansion and query refinement.

Query expansion: the initial query is expanded with two type of generated terms:

- The terms that represent the state attributes, from UML state diagram, for the current task A_* (denoted a_{S1}, a_{S2}, …,a_{Si}) One state attribute for each task state.
- The query-relevant attributes from the ontological user profile with its values. (<attribute a_{u1}, value>, <attribute a_{u2}, value>, …,<attribute a_{uj}, value>)

Query refinement: Query refinement is the process of transforming a query into a new query SRQ that more accurately reflects the user's information need. Sometimes irrelevant attributes may be present in the selected user profile concepts. In order to keep only the relevant user profile attributes for the current task state S_i, we compare between these generated attributes and the current state attributes, next we exclude from the generated user profile attributes these non similar with the state attributes. We must also exclude the duplicated terms if they exist in the resulting SRQ.

Another method for filtering the previous terms is by asking the user to choose the relevant terms before adding them to the query.

Finally SRQ is built according to the syntax required by the used search engine in order to submit the query SRQ and to provide back results to the user.

Let *q* an initial query which is composed of many terms $\{t_1, t_2, ..., t_n\}$ and related to the task at hand. The state reformulated query in the task state S_i and for a specific user profile P_j is: $S_iRQ<Q,P_j,S_i>$, The relevant results D_i in the states S_i are produced by applying $S_iRQ<Q,P_j,S_i>$ on an information retrieval system. We expect that the results D_i in the task state S_i are more relevant than the normal results produced by using the initial query *q* in S_i, to check that an experimental study will be performed.

IV. System Architecture

Fig. 3 presents the system architecture. It combines the several models described in the previous section: the task model, the contextual state model, the ontological user profile model and the SRQ model.

V. Experimental Study

Here we first suppose that the queries we are considering are related to some current task at hand and secondly, the tasks are modeled by UML state diagrams. We can show that our system works depending on the following practical consequent steps:

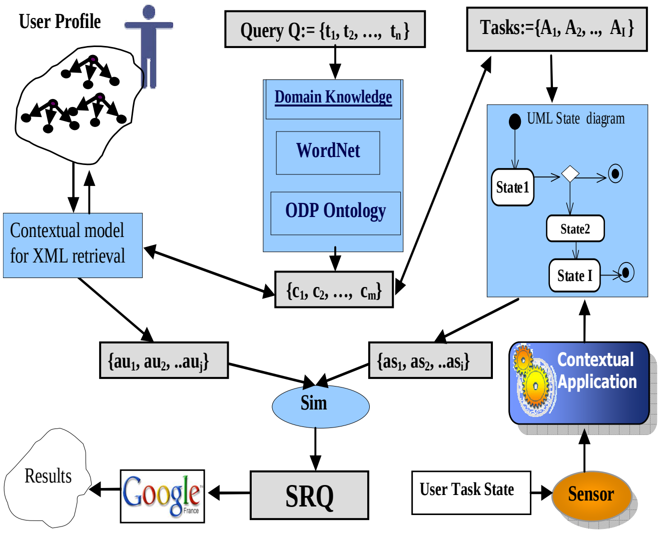


Figure 3. System architecture.

When the user submits his query in our platform, the system will detect the user current task (described in task model, section III paragraph B) as the first step. Next, the UML state diagram for this task is retrieved (section III paragraph C). The system then uses the attributes associated with each state (in UML) and the user profile attributes for producing the relevant terms (methodology section III paragraph E). The irrelevant terms are excluded (The query refinement). Finally, the reformulated query denoted SRQ is submitted to Google to retrieve the relevant results.

For instance, Let us consider the query *q*= {Buy Laptop}, the task assigned to the user query *q* is: "Shopping and Selling". The contextual state model allows the proposition of several task states that are represented in UML state diagram as shown in the fig.4. For this task the system can produce the following SRQ:

- S_1 (Information about laptop models): S_1RQ: {"laptop"+ information}.
- S_2 (model choice): S_2RQ:{"laptop"+ HP OR Asus}.
- S_3 (comparing prices): S_3RQ:{ "laptop"+ price OR Inexpensive}.
- S_4 (choosing a computer shop): S_4RQ: {"laptop"+ address OR London}.

Table 2 presents the state reformulated queries SRQ for the query *q* and their relevance score using the first 20 retrieval results of Google. For example, at the first task state S_1 which is "general information about laptop models", there are 11 relevant results of 20 retrieved by Google using the user query *q* without reformulation, while there are 14 relevant results of 20 using the SRQ.

The evaluation of such systems is complicated due to the dynamic aspect of the system environment. So, we performed two manual evaluations, one to evaluate the detected task and another to evaluate the SRQ (State Reformulated Queries):

We asked 10 different users to submit 3 queries (for doing different tasks) the system then detects the task for each query. Next the users are asked if the queries were their tasks or not. We then got nearly 21 out of 30 positive responses (70%).

To evaluate the SRQ queries we asked the 10 users to submit different queries and we applied each one to the Google search engine at the different states of the task which was detected by our task model. We reformulated these queries by adding the relevant terms and then we reapplied them at the states using the same search engine. We compared the first 20 retrieval results produced in the two cases (by queries *q* and queries SRQ).

Results: we calculated the average number of relevant pages by queries *q* and SRQ on the first 20 results (P@20). We noticed that the precision of the relevant results using the initial query *q* is 0.17 and 0.59, respectively, by using SRQ queries which were reformulated depending on the current task state and user profile.

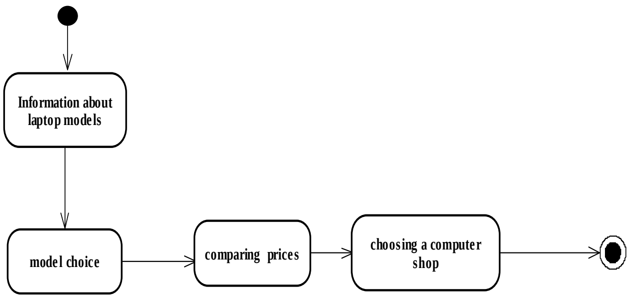


Figure 4. shows an example of a task that is modeled by UML state diagram.

TABLE II. THE STATE REFORMULATED QUERIES FOR THE QUERY Q.

Query	Q				S_1RQ	S_2RQ	S_3RQ	S_4RQ
Terms	Buy laptop				"laptop"+ information	"laptop" + HP OR Asus	"laptop" + price OR Inexpens ive	"laptop" + address OR London
P@20	S_1	S_2	S_3	S_4	14	15	8	7
	11	2	4	1				

VI. CONCLUSIONS

In this paper, we have proposed a hybrid method to reformulate user queries depending on a dynamic ontological user profile and user context for producing State Reformulated Queries (SRQ). The user context is considered as the actual state of the task that he is undertaking when the information retrieval process is performed. We have constructed a general architecture that combines several models for query expansion: the task model, the contextual model, the user profile retrieval model and SRQ model. We exploit both a semantic knowledge (ODP Ontology) and a linguistic knowledge (WordNet) to learn user's task, and we exploit a UML states diagram for this task to learn user current state. We have also constructed a new general language model for query expansion including the contextual factors and user profile. We have illustrated on an experimental study that the results obtained by SRQ queries are more relevant than those obtained with the initial user queries in the same task state. As a future work, we plan to evaluate this method by creating a test collection.

REFERENCES

[1] Micarelli, A., Gasparetti, F., Sciarrone, F., and Gauch, S.: Personalized Search on the World Wide Web. P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.), In: The Adaptive Web, LNCS 4321, pp. 195–230, Berlin, 2007.

[2] Liiv I., Tammet T., Ruotsalo, T., and Kuusik A.: Personalized Context-Aware Recommendations in SMARTMUSEUM Combining Semantics with Statistics. In: 2009 Third International Conference on Advances in Semantic Processing, Malta 2009.

[3] Kofod-Petersen, A. and Cassens, J.: Using Activity Theory to Model Context Awarenessa, In: American Association for Artificial Intelligence, Berlin, 2006.

[4] Conesa, J., Storey, V.C., and Sugumaran, V.: Using Semantic Knowledge to Improve Web Query Processing, In: NLDB 2006, pp. 106 – 117, Springer-Verlag Berlin, 2006.

[5] Chirita, P. A, Nejdl, W., Paiu, R., and Kohlschutter, Ch.: Using ODP Metadata to Personalize Search. In: Proc. of the 28th Annual Int'l ACM SIGIR Conf., 178–185, Brazil, 2005.

[6] Koutrika, G., and Ioannidis, Y. E.: Personalization of Queries in Database Systems. In: Proceedings of the 20th International Conference on Data Engineering, USA, 2004.

[7] Vidal, M.E., Raschid, L., Marquez, N., Cardenas, M., and Wu, Y.: Query Rewriting in the Semantic Web. In: Proceedings of the 22nd International Conference on Data Engineering Workshops, ICDEW, USA, 2006.

[8] Baeza-Yates, R., Hurtado, C., and Mendoza, M.: Query recommendation using query logs in search engines. In EDBT '04, 588-596, 2004

[9] Wakaki, H., Masada, T., Takasu, A., and Adachi, J.: A New Measure for Query Disambiguation Using Term Co-occurrences. In: Lecture Notes Computer Science, 2006, NUMB 4224, pages 904-911.

[10] Navigli, R. and Velardi, P.: An Analysis of Ontology-based Query Expansion Strategies. Workshop on Adaptive Text Extraction and Mining at the 14th European Conference on Machine Learning, Cavtat-Dubrovnik, Croatia (2003).

[11] Bhogal, J., Macfarlane, A., and Smith, P.: A review of ontology based query expansion, Information Processing and Management. In: an International Journal, v.43 n.4, p.866-886, July, 2007, doi: 10.1016/j.ipm.2006.09.003.

[12] Sieg, A., Mobasher, B., and Burke, R.: Representing context in web search with ontological user profiles. In: Proceedings of the Sixth International Conference on Modeling and Using Context, Roskilde, Denmark, August 2007.

[13] Asfari, O.: Modèle de recherche contextuelle orientée contenu pour un corpus de documents XML. In: CORIA 2008: 377-384, France.

[14] Asfari, O., Doan, B. L., Bourda, Y., and Sansonnet, J.-P.: Personalized access to information by query reformulation based on the state of the current task and user profile, In: The Third International Conference on Advances in Semantic Processing, SEMAPRO, Malta, 2009.

[15] Bouchard, H. and Nie, J.Y.: Modèles de langue appliqués à la recherche d'information contextuelle, In: Conf. en Recherche d'Information et Applications (CORIA), Lyon, 2006.

[16] W. White, R. and Kelly, D.: A Study on the Effects of Personalization and Task Information on Implicit Feedback Performance, In: CIKM'06, 2006, USA.

Modeling Action Game Domains Using Latent Semantic Analysis

Katia Lida Kermanidis
Department of Informatics
Ionian University
Corfu, Greece
kerman@ionio.gr

Kostas Anagnostou
Department of Informatics
Ionian University
Corfu, Greece
kostasan@ionio.gr

***Abstract*—Player modeling has attracted the interest of game designers recently, as a personalized game offers more satisfaction. In this paper we propose modeling the semantic space of the action game SpaceDebris, in order to identify semantic similarities between players. To this end we employ Latent Semantic Analysis and attempt to identify latent underlying semantic information governing the various gaming styles. The several challenging research issues that arise when attempting to apply Latent Semantic Analysis to non-textual data, that describe a complex dynamic problem space, are addressed, and the framework of the experimental setup is described.**

player modeling, action games, latent semantic analysis, knowledge representation, semantic similarity

I. INTRODUCTION

Representing the knowledge of a specific domain, i.e. identifying the concepts that carry units of meaning related to it (domain “words”), as well as the semantic relations governing those concepts, is a wide and popular research area. Modeling domain knowledge is essential for developing expert systems, for intelligent prediction and decision making, for intelligent tutoring, user modeling, complex problem solving, reasoning etc. Mastering the semantics of a domain is to learn the “language” of the domain [12], i.e. to become exposed to various sequences of domain “words” in numerous contexts. This is similar to the way a foreign language learner learns vocabulary usage by reading, listening to, and writing texts in that language.

There are two possible ways for supplying domain knowledge [12]: by hand, making use of domain experts’ know-how, and automatically, by deriving the semantics from large corpora of “word” sequences. The first approach is more accurate, but domain-dependent, while the second is useful when no hand-crafted knowledge is available.

A widely used method for representing domain knowledge by statistical analysis of word usage is Latent Semantic Analysis (LSA). LSA is adopted from the field of Information Retrieval [11] and improves retrieval performance by taking into account automatically detected polysemy and synonymy relations between words. LSA identifies these underlying semantic relations by exploiting the occurrence statistics of the words throughout the document collection: by reducing the dimensionality of the initial term-document matrix[1], hidden semantic similarities between words, between documents, and between words and documents surface, linking together words that may not even appear in the same document, or documents that may not share any common words.

LSA has been applied with significant success to other domains, like essay assessment in language learning [8], intelligent tutoring [7], text cohesion measurement [13], summary evaluation [18], text categorization [14]. Although all previously mentioned LSA applications have been performed on text corpora, some approaches have proposed its use in different non-textual knowledge domains like board game player modeling [22], complex problem solving [15], gene function prediction [3][4][5], web navigation behavior prediction [21], collaborative filtering [10], semantic description of images [2].

The present work proposes the application of LSA to a new domain, namely digital action games, in order to identify similarities among the playing techniques of various players. Thereby, the players’ profile can be constructed, and games can be adapted to individual players’ needs and preferences, offering more satisfaction.

Action games have properties that resemble those of complex dynamic environments: causality relations (actions or decisions often affect subsequent actions or decisions), time dependence (the environmental circumstances that affect actions and decisions vary over time), and latent, implicit relations between domain properties that are not straightforward. Identifying the domain vocabulary, as well as well-formed sequences of “words” that constitute complete descriptions of actions or context conditions is of significant research interest.

Throughout the remainder of the paper we will address the research challenges that emerge when attempting to represent the semantics governing the SpaceDebris action

[1] The matrix with rows representing index terms and columns representing documents, and each cell contains the number of occurrences of a term in a document.

shooting game [1]. The proposed use of the representation is player modeling: unsupervised grouping of players with similar gaming manners. Section 2 provides a bibliographic review of player modeling and categorization. Section 3 presents the basic properties of Latent Semantic Analysis, section 4 introduces the action game SpaceDebris, and finally section 5 describes the cognitive modeling process of the game domain, as well as its use for modeling players.

II. Player Profiling

Several game designers have recently been shifting their focus to the player rather than the game itself. Numerous attempts have been made to identify the gaming technique of each player (e.g. (in)experienced, aggressive, tactical, action player), aiming to adapt the game features to his individual preferences and needs. By personalizing the features of the game, the designer hopes to provide increased satisfaction and entertainment.

Player modeling has been achieved within an interactive storytelling game and the use of machine learning techniques [20][16], by estimating the statistical behavior (distribution) of player actions [19], by using graphical knowledge representation schemata like influence diagrams [17] and Bayesian networks [9]. Further references to player modeling can be found in [6]. In [1] SpaceDebris players are grouped into two clusters, using unsupervised learning, according to their playing style (aggressive or tactical).

Unlike previous approaches that either assign one of a set of predefined profiles to a player, or explore explicit actions and decisions made by the player, the present work proposes a knowledge model that attempts to

- identify the vocabulary of the game domain,
- represent complicated game states (action game states are hard to represent, as their definition is not straightforward like in board games), and
- detect hidden, underlying semantic relations between decisions made and actions taken and their context, as well as among domain "words".

III. Latent Semantic Analysis

As mentioned earlier, LSA is a mathematical/statistical method initially proposed for reducing the size of the term-document matrix in information retrieval applications, as the number of lexicon entries may reach several thousand, and the document collection may contain tens of thousands of documents or more. LSA achieves dimensionality reduction through Singular Value Decomposition (SVD) of the term-document matrix. SVD decomposes the initial matrix *A* into a product of three matrices and "transfers" matrix *A* into a new semantic space:

$$A = T\,S\,D^T \qquad (1)$$

T is the matrix with rows the lexicon terms, and columns the dimensions of the new semantic space. The columns of *D* represent the initial documents and its rows the new dimensions, while *S* is a diagonal matrix containing the singular values of *A*. Multiplication of the three matrices will reconstruct the initial matrix. The product can be computed in such a way that the singular values are positioned in S in descending order. The smaller the singular value, the less it affects the product outcome. By maintaining only the first few of the singular values and setting the remaining ones to zero, and calculating the resulting product, the initial matrix may be approximated as a least-squares best fit. The dimensions of the new matrix are reduced and equal to the number of selected singular values.

As an interesting side effect, dimensionality reduction reduces or increases the frequency of words in certain documents, or may even set the occurrence of words to higher than zero for documents that they initially did not appear in. Thereby semantic relations between words and documents are revealed that were not apparent at first (latent). It needs to be noted that LSA is fully automatic, i.e. the latent semantic relations are learned in an unsupervised manner. Another significant property is that LSA does not take into account the ordering of words within their context; documents are considered "bags of words". Extensive information on LSA can be found in [11].

IV. SpaceDebris

The videogame used for the purposes of data collection is based on SpaceDebris [1]. The action takes place within the confines of a single screen, with alien ships scrolling downwards. There are two types of enemy spaceships, the carrier which is slow and can withstand more laser blasts, and a fighter which is fast and easier to destroy. The player wins when he has successfully withstood the enemy ship waves for a predetermined time. The game environment is littered with floating asteroids which in their default state do not interact (i.e. collide) with any of the game spaceships. In order to do so, an asteroid has to be "energized" (hit by player weapon). Also floating are shield and life power-ups which the user can use to replenish his ship's shield and remaining lives. The player's ship is equipped with a laser cannon which she can use to shoot alien ships. The laser canon is weak and about 4-5 successful shots are required to destroy an enemy ship (except for the boss which requires many more). The laser can also be used to "energize" an asteroid and guide it to destroy an enemy ship.

V. Modeling SpaceDebris

Several research challenges need to be addressed when attempting to model the domain of an action game like SpaceDebris using LSA.

A. Vocabulary Identification

In board-like games, like tic-tac-toe or chess, domain "words" are easy to identify. Boards may be viewed as grids of cells and each cell state (e.g. "X", "O" or empty in tic-tac-toe) constitutes a "word" [12]. In action video games "words" are harder to identify. Should they represent player actions, enemy actions, the state of the context, scoring results, spare lives or ammunition, time parameters? In the firefighting microworld of [15] "words" are actions like appliance moves, or water drops. The definition of a game "word" depends on the intended use of the model. If the intended use is behavior prediction, a “word” needs to model a player’s action, as the player’s sequence of actions (in a given context) defines his behavior.

In the present work, two approaches to representing “words” are considered. In the first approach, the game terrain is considered a grid, and the concatenation of the states of each cell in the grid constitutes a “word”. The state of each cell is determined by several factors, depending on the state of each game entity. For example a cell might be empty, it might contain an asteroid, it might contain an “energized” asteroid. It might also contain the player’s ship, the player’s ship firing a laser, the player’s ship being hit by a laser. A cell might also be in state that combines a number of states such as those described. Player or enemy actions are modeled implicitly through the related cell states. Further out-of-the-grid (non-spatial) information, like score, spare lives, spare shields, is modeled separately and each of these features is concatenated to the cell states to constitute a complete “word”. The cell size is of importance, as it affects the level of granularity. The smaller the cell size is, the more “generic” the “words” are. We will experiment with grid sizes 11x8 and 12x6, the first corresponding to the player ship’s size and the second to the largest enemy ship size, with a screen resolution of 1024x768 pixels. Vocabulary size using this representation of approximately 24 cell states reaches 2212 with a grid size of 11x8 and 1728 with grid size of 12x6. Vocabulary size is important, as too many “words” may result to too few cooccurrences and LSA will not work. A too small vocabulary may lead to too few similarities and, again, the method will not work [12]. Optimal vocabulary size is an open research issue and depends on the domain.

The second approach is more “holistic” and resembles that of [15]. Each “word” represents a player action, like *move to a location* or *fire*. However, unlike [15], each action in a “word” is accompanied by a concatenation of features that represent the state of the context in which the action took place. These features are

- the number of enemies very close to the player
- the number of enemies close to the player
- the total number of enemies on the screen
- the number of player lasers fired
- the number of enemy lasers fired
- the position of the player
- the number of life and shield upgrades performed
- the number of hit asteroids
- the number of visible asteroids
- the number of hit enemy ships
- the score value
- the number of available life upgrades
- the number of shields available to the player

“Word” examples using an NxM grid (ex. 1) and the “holistic” (ex. 2) approach are shown below. The first part (up to X_{NM}) of the string in ex. 1 consists of tokens, each token stands for one cell state (tokens are concatenated together with underscores). We use 16bit numbers, to denote the presence (1 or 0) of one of the 9 game entities (player, 2 types of enemies, 3 types of lasers, 2 types of upgrades, asteroid). The last three tokens encode out-of-the-grid information, i.e. the score, the number of spare lives and spare shields respectively. In ex. 2 the first token is the player’s action (the player moves to location with coordinates (-286, -133)). Each of the following concatenated tokens is a value for each of the features listed above (e.g. 1 enemy is very close, 3 are close, there are 9 enemies on-screen, player has fired a laser, enemies have fired 3 lasers etc.).

2_1_0_..._X_{NM}_1000_3_100 (ex. 1)
move-286-133_1_3_9_1_3 _...._X (ex. 2)

The “grid” representation takes into account long-distance semantic dependencies, i.e. the semantics of each cell (no matter how distant) participates in the domain knowledge. The “holistic” representation detects causality relations between the environment and the player’s reaction to it in a more straightforward way.

B. *Game Session Representation*

Game sessions play the role of documents in Information Retrieval. As documents are sequences of words that convey a specific meaning and are considered to satisfy a certain information need, game sessions are well-formed sequences of “words” in the game domain. Each “word” constitutes a complete description of a player’s action or of a description of the context (game environment) at a given moment.

One way to represent a game session is to take a sample of the game state at constant pre-defined time intervals (e.g. every 500 msecs) and register the sequence of “words” (“words” are defined using either the grid or the holistic approach) that describe the sample. Each sample represents a game state at the specified time point. The duration of the sampling time interval is very important. Small intervals may lead to consecutive states that are semantically identical (i.e. the player has not had enough time to make a decision or act, or the state of the context has not changed). Long intervals may lead to the loss of semantic information (i.e. player’s actions that occurred between the samples may be missed). We will experiment with various interval sizes in order to find the “optimal” sampling rate.

Another way to represent game sessions is through sampling events that are dynamically triggered by player’s actions. Every time the player acts, a game state sample is taken, and the player’s action and game context are recorded.

C. *Reduction Rate*

The rows of the resulting term-document matrix represent the “words”, and the columns represent game sessions. Each cell contains the frequency of occurrence of the “word” in the row in the column session. Applying LSA to the matrix, another research question arises: What is the optimal number of singular values that should be maintained? In Information Retrieval the number of dimensions of the latent semantic space is usually between 100 and 300 [12]. More research work needs to be done in order to determine the appropriate number of dimensions when it comes to non-textual domains. Our proposal includes the experimentation with various dimension numbers and the research of their impact on modeling performance.

D. Experimental Setup for Measuring Semantic Similarity

As mentioned earlier, the extracted model will be used for identifying similar gaming techniques among players. A group of players will play the game for a given time frame. Players will at first be asked to familiarize themselves with the game by playing off the record for 4-5 minutes. After this introductory phase, game sessions will be recorded for every player. Each game session lasts an average of 3 minutes, and players will be asked to complete a specific number of games. The number of games needed for successfully identifying the player's gaming style will be experimentally explored. Each game session will constitute a feature vector, which is formed by the set of "words" representing it. Feature vectors both before and after LSA will be stored for comparative analysis of results.

To identify similar gaming techniques, the distance between vectors needs to be computed. Though several distance metrics have been experimented with, pairwise cosine similarity is the most popular measure [12]. Cosine similarity will link the most semantically similar vectors together, forming unsupervised clusters of similar gaming techniques. Clustering evaluation may be performed in two ways. Players may be asked to answer a short questionnaire before playing, where they will characterize their individual gaming style, choosing one or more from a set of pre-defined styles. Another way is to ask a game expert to identify the style of each individual player by looking at his actions and decisions throughout the game sessions. The matching degree of the cosine similarity and the expert's decision (and/or the player's questionnaire answers) will be measured before and after applying LSA, for detecting its impact.

VI. CONCLUSION

In this paper we have described a proposal for modeling the semantic space of a complex non-textual problem, i.e. an action game, using LSA. While the application of LSA to textual data is fairly straightforward, several research issues arise when the data involved are not textual, but represent players' actions and environmental (contextual) conditions. These research issues have been addressed and an experimental setup has been proposed for the novel use of the extracted model to player modeling.

REFERENCES

[1] K. Anagnostou and M. Maragoudakis, "Data mining for player modeling in videogames", Proc. of the Panhellenic Conference on Informatics (PCI), Corfu, Greece, 2009.

[2] R. Basili, R. Petitti, and D. Saracino, "LSA-based automatic acquisition of semantic image descriptions", Proc. of the Conference on Semantics and Digital Media Technologies, LNCS, vol. 4816, 2007, pp. 41-55.

[3] B. Done, P. Khatri, A. Done, and S. Draghici, "Predicting novel human gene ontology annotations using semantic analysis", IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 7(1), January-March 2010. pp. 91-99.

[4] Q. Dong, X. Wang, and L. Lin, "Application of latent semantic analysis to protein remote homology detection", Bioinformatics, vol. 22(3), Oxford University Press, 2006, pp. 285-290, doi: 10.1093/bioinformatics/bti801.

[5] M. Ganapathiraju, N. Balakrishnan, R. Reddy, and J. Klein-Seetharaman, "Computational biology and language", Ambient Intelligence for Scientific Discovery, LNAI, vol. 3345, 2005, pp. 25-47.

[6] B. Geisler, "An empirical study of machine learning algorithms applied to modeling player behavior in a first person shooter video game", MSc Thesis, 2002, University of Wisconsin-Madison.

[7] A. C. Graesser, P. Penumatsa, M. Ventura, Z. Cai, and X. Hu, "Using LSA in AutoTutor: Learning through mixed-initiative dialogue in natural language", Handbook of Latent Semantic Analysis, T. Landauer, D. McNamara, S. Dennis, W. Kintsch Eds, 2007.

[8] D. T. Haley, P. Thomas, A. de Roeck, and M. Petre, "A research taxonomy for latent semantic analysis-based educational applications", Proc. of the Conf. on Recent Advances in Natural Language Processing, Borovets, Bulgaria, 2005.

[9] S. He, J. Du, H. Chen, J. Meng and Q. Zhu, "Strategy-based player modeling during interactive entertainment sessions by using Bayesian classification", Proc. of the 4th International Conference on Natural Computation (ICNC), November 2008, pp.255-261.

[10] T. Hofmann, "Latent semantic models for collaborative filtering", ACM Trans. on Information Systems, vol. 22(1), January 2004, pp. 89-115.

[11] T. Landauer, P. Foltz, and D. Laham, "An introduction to latent semantic analysis", Discourse Processes, vol. 25, 1998, pp. 259-284.

[12] B. Lemaire, "Models of high-dimensional semantic spaces", Proc. 4th International Workshop on Multistrategy Learning (MSL), June 1998.

[13] P. M. McCarthy, S. Briner, V. Rus, and D. McNamara, "Textual signatures: identifying text-types using latent semantic analysis to measure the cohesion of text structures", In A. Kao and S. Poteet (Eds), Natural Language Processing and Text Mining, March 2007, pp. 107-122, doi: 10.1007/978-1-84628-754-1_7

[14] P. Nakov, E. Valchanova, and G. Angelova, "Towards deeper understanding of the LSA performance", Proc. of the Conf. on Recent Advances in Natural Language Processing, 2003.

[15] J. F. Quesada, W. Kintsch, and E. Gomez, "A computational theory of complex problem solving using the vector space model (part I): latent semantic analysis, through the path of thousands of ants", In J. J. Cañas (Ed.), Cognitive research with Microworlds, Granada, Spain, 2001, pp. 117-131.

[16] D. Roberts, M. Riedl, and C. Isbell, "Opportunities for machine learning to impact interactive narrative", Workshop on Machine Learning and Games at NIPS, 2007.

[17] G. Shahine and B. Banerjee, "Player modeling using knowledge transfer", Proc. of EUROSIS GAMEON-NA Conference, Gainesville, Florida, 2007, pp. 82–89.

[18] J. Steinberger and K Jezek, "Using latent semantic analysis in text summarization and summary evaluation", Proc. of Conf. on Information Systems, Implementation and Modeling (ISIM), 2004, pp. 93-100.

[19] R. Thawonmas and J. Ho, "Classification of online game players using action transition probability and Kullback Leibler entropy", Journal of Advanced Computational Intelligence and Intelligent Infromatics, Special issue on Advances in Intelligent Data Processing, vol. 11(3), 2007, pp. 319-326.

[20] D. Thue, V. Bulitko, M. Spetch, and E. Wasylishen, "Interactive storytelling: a player modelling approach", Proc. of the 3rd Conf. on Artificial Intelligence and Interactive Digital Entertainment (AIIDE), Stanford, California, USA, June 2007, pp. 43-48.

[21] H. van Oostendorp and I. Juvina, "Using a cognitive model to generate web navigation support", International Journal of Human-Computer Studies, Elsevier, vol. 65(10), October 2007, pp. 887-897, doi:10.1016/j.ijhcs.2007.06.004.

[22] V. Zampa and B. Lemaire, "Latent semantic analysis for user modeling", Journal of Intelligent Information Systems, Kluwer Academic Publishers, vol. 18, 2002, pp. 15-30.

Ontological CAD Data Interoperability Framework

Luis Enrique Ramos García*
**University of Bremen*
Cartesium, Enrique-Schmidt-Strasse. Bremen, Germany
s_7dns7r@uni-bremen.de
**Universidad Nacional Abierta*
lramos@una.edu.ve

***Abstract*—Computer Aided Design (CAD) has become one of the fundamental activities in the modern industry. Nowadays several products are developed and modeled using this technology. Nevertheless, extracting product features from these kind of files to use them in production processes and parametric data exchange among heterogeneous CAD systems are still difficult to achieve. This work aims to propose OWL as CAD Data Exchange Format, giving the possibility for the addition of more descriptive information of products and processes in one self-content and self-descriptive file. With this CAD - OWL integration the feature extraction is facilitated, because this CAD - OWL model becomes a Knowledge Base and the reasoning tools of the Semantic Web become available. In this work a standard CAD file was mapped into the Web Ontology Language (OWL) and visualized using the Protégé API's architecture in order to deal with such problems.**

***Keywords*-Ontology; Web Ontology Language (OWL); Computer Aided Design(CAD); Protégé.**

I. Introduction

Computer Aided Design has been an important approach for designing of mechanical parts since the beginning of the 1970's. This technology has a fundamental role in the industrial processes of manufacturing. Software CAD tools such as AutoCAD®[1], Pro Enginer®[2], Free CAD [3] and others are used nowadays at the beginning of these processes, specifically in the products design phase (see Fig. 1). After this design is ended, the production process continues with the production planning phase, where among other things, tasks as determining manufacturing, getting valid raw material suppliers, calculating costs, time, quantity of production, selecting the kind of machines needed, sequence of operation, etc., take place [4]. In the manufacturing phase this design becomes a product. Cause these activities are highly time consuming, and repetitives, there have been efforts to make automatic extraction of information from CAD files using parser computer programs in order to generate Automatic Production Planing and Manufacturing in two research and application areas known as Computer Aided Process Planing (CAPP) and Computer Aided Manufacturing (CAM) [5]. This automated interaction and evolution has been limited cause by the semantic weakness of the CAD standards for these objectives, which in fact, are not designed for this kind of interaction, but for the representation of geometrical elements, such as lines, circles, surfaces, solids of revolution and so on [6]. Besides, in the process planing and production phases it is necessary interact with expert knowledge, which is difficult to represent and reuse [7]. To deal with this limitations and in order to facilitate the interaction with a CAD design we propose the ontological approach as a way to represent expert knowledge and the semantic web as a platform for CAD - CAPP - CAM Processes Interaction Automation. In this article a first experience is presented, where a CAD data exchange format called DXF was used to classify a two dimensional design in a set of classes and instance in order to populate a geometry ontology. By making queries to the ontology this design was rebuild for visualization using Protégé 3.4.4 and Java 2D as a prototyping platform.

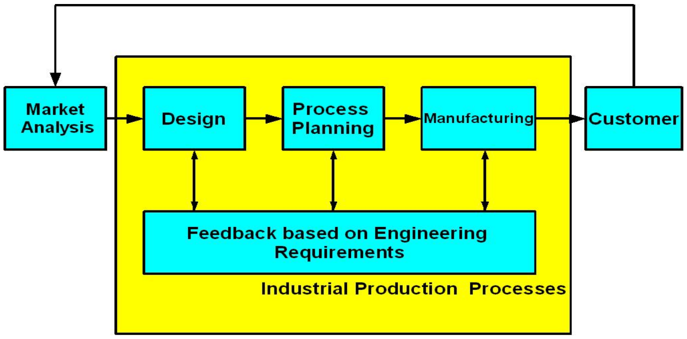

Market Analysis
Design
Process Planning
Manufacturing
Customer
Feedback based on Engineering Requirements
Industrial Production Processes

Figure 1. Industrial Production Process

II. Previous Works

Works related with the extraction of features from a CAD file have been reported since the 1980s. Henderson and Anderson [8] used PROLOG, a rules-based language to express a set of necessary and sufficient conditions to classify features. They found that PROLOG had limitations to handle trigonometric functions required to deal with general angular relationships. In their work, even, a not optimized sequence of production was generated, mechanical parts were manufactured automatically. Sam Lazaro *et al* [9] developed an intelligent system to help designers in developing metal sheet parts, using an object based Knowledge Base System development package called NEXTPERT Object. Design rules were represented in this system as a set of IF <list of conditions>THEN <hypotesis><list of actions>. Although the work of [9] did not deal with the generation of production sequences, it is an example of the use of Knowledge Bases to store expert knowledge and its re-utilization to alert designer engineers when a design rule was violated. Soman *et al* [11], developed a system using C++, in which rules were used in the automatic production of a design for a sheet metal part. Here the application of a certain group of rules was controlled by a list of conditional loops, facilitating the automation of the design production process. In this work an estimation of manufacturing time and cost prediction was reported, although it was not described any optimization module or process.

The critics made to these methodologies, based on rules, are the rules themselves, because the designer of such system needs to develop rules enough explicit for each case, some languages as C++ are not intended to make rules, some rule languages as PROLOG has limitation for certain mathematical operations, which limits these programs for this kind of application.

In recent years, there has been a movement toward the utilization of the ontological approach in engineering applications for the representation of CAD models to capture feature semantics and to use such model among different system maintaining the designer's intent. Ghafour *et al* [12], presented an architecture for a Data Exchange among different CAD software tools, in where ontologies are proposed to represent the terminologies of some commercial CAD software tools, and a main ontology would serve as a Common Design Feature Ontology. He proposed to write and store ontologies on each CAD system using the Web Ontology Language (OWL) a W3C standard, generating an ontology of such systems. These ontologies have to be mapped in a Common Design Ontology to make them interoperable among different software applications. Similarly, Odd and Vasilakis [13] proposed an ontology of CAD model information, this proposal is described as an introduction to ontologies and shapes representation. It deals with the Standard for the Exchange of Product data (STEP) [14] similar to [12], presenting a taxonomy of terminologies included in the STEP standard. Grüninger and Delaval [15], proposed a set of ontologies related with shapes, shape cutting and cutting process. Although this work is not related with direct feature extraction from a standard CAD file, his proposal aimed to deal with the lack of shareability and reusability related with the ruled based feature extraction approachs. He proposed his ontologies using First Order Logic to make a mathematical generalization and verification.

III. The Drawing Exchange Format (DXF)

The Drawing Exchange Format is a *de facto* standard for the interchange of CAD data, which facilitates reading a CAD design previously deployed using software tools as [1]. A detailed explanation of this standard can be found in the DXF Reference Manual [16]. This standard defines geometric primitives as entities such as LINE, CIRCLE, ARC and ELLIPSE. For them, a group of codes is specified indicating what type of data value or feature follows. Besides, from a DXF file is possible to extract descriptions of text, surfaces, color, texture, but the information about solids is not accessible [17], that limits the exchange of data and information with other CAD applications. Nevertheless this work is intended to identify primitive as LINE, CIRCLE, ARC and ELLIPSE stored in a DXF formatted file to populate an ontology and store it as an OWL file. In Table I a description of codes for the primitive LINE is shown, there can be seen that this ”entity” is identified with AcDbLine, after that, features are presented, start point and end point values in X, Y, and Z axis are given for this LINE. This definition is similarly described by the DXF specification for CIRCLE, ARC and ELLIPSE, but considering their geometric features.

Table I
Description of an entity as it appears in a DXF file

DXF file code	Meaning
100	Sub Class
AcDbLine	Name of entity
10	Start point in X
20.83	Value of start point in X
20	Start point in Y
49.27	Value of start point in Y
30	Start point in Z
0.0	Value of start point in Z
11	End point in X
115.44	Value of end point in X
21	End point in Y
91.06	Value of end point in Y
31	End point in Z
0.0	Value of end point in Z
0	End of entity
ENDSEC	End of sequence

IV. Ontologies as Representation of Standards

Ontologies are defined as a specification of a share conceptualization [18]. An ontology includes concepts and relations, it has to be general enough to represent the sharable knowledge in an specific domain. For CAD systems, a domain ontology should represent the common elements of the most accepted and used CAD standard formats. Based on the quantity of scientific and technical references and CAD software tools that we have reviewed until now, we consider that those formats are DXF, IGES and STEP. So, our proposed ontology consist of a group of geometry

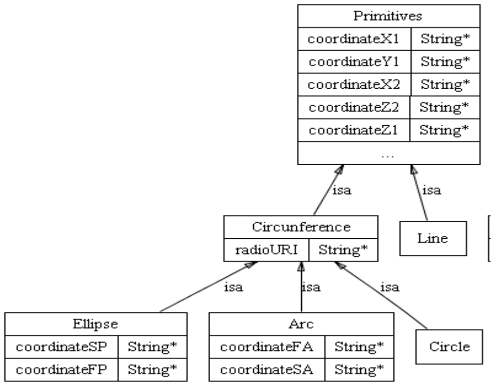
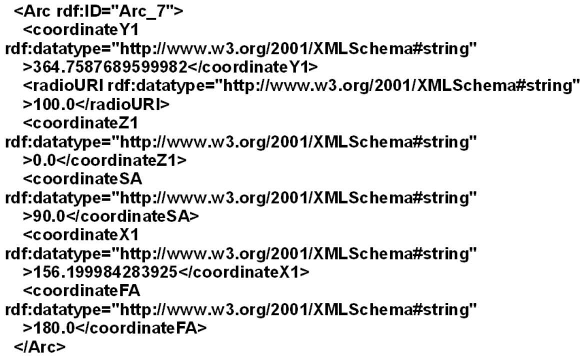


Figure 2. Ontology representation for CAD exchange



```
  <Arc rdf:ID="Arc_7">
    <coordinateY1
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >364.7587689599982</coordinateY1>
    <radioURI rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >100.0</radioURI>
    <coordinateZ1
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >0.0</coordinateZ1>
    <coordinateSA
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >90.0</coordinateSA>
    <coordinateX1
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >156.199984283925</coordinateX1>
    <coordinateFA
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >180.0</coordinateFA>
  </Arc>
```

Figure 3. Part of a DXF file represented in OWL

primitives defined as classes and its features (Data type properties), which are shown in Figure 2. These classes were defined in a Java Program using the API of Protégé 3.4.4.

After declaring that ontology, the class scanner of Java was used to read each line of a DXF file getting each one of the specific primitives and store them as instance of the ontology in their specific class, the features of the primitives are identified using the code defined for the instance on the DXF specification and stored in the ontology (model). After reading the whole DXF file, an OWL file is generated. On Figure 3, a resulting OWL file of a CAD model can be seen, this partial view shows an instance Arc_7 with its features (Datatype properties)

V. Implementation and Prototyping

Our first implementation, a DXF OWL API importer, was tested with two and three dimensional CAD models, and the extraction of primitives was successful, including all features of each instance in the files. But, as the DXF file belongs to a graphical representation, after we had the OWL Model it was decided to develop a second API in order to make possible the visualization of CAD - OWL models in Protégé as it can be seen in a CAD software tool. In Figure 4 a CAD model is presented, which was exchanged to OWL and can be seen in Fig. 5. This CAD viewer is limited to two dimensions, because our objective was not to make another CAD tool, but this viewer facilitates the human work for the verification of the correct exchange of the model.

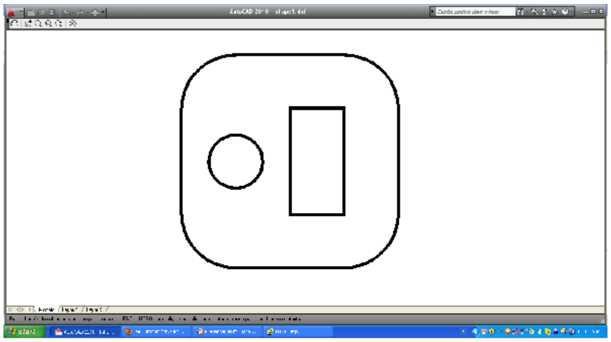


Figure 4. Shape modeled in AutoCAD 2010

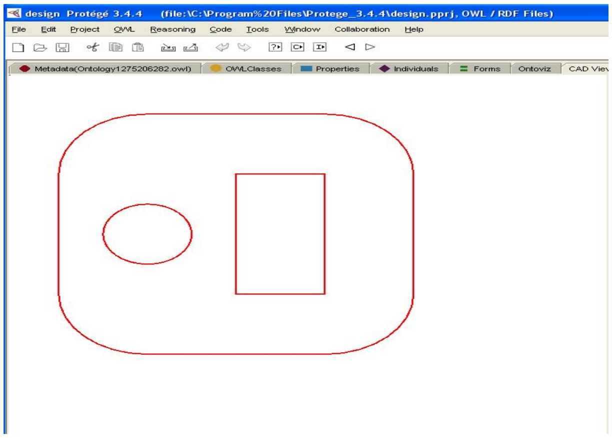

design Protégé 3.4.4 (file:\C:\Program%20Files\Protege_3.4.4\design.pprj, OWL / RDF Files)
File
Edit
Project
OWL
Reasoning
Code
Tools
Window
Collaboration
Help
Metadata(Ontology1275206282.owl)
OWLClasses
Properties
Individuals
Forms
Ontoviz
CAD Viev

Figure 5. Shape modeled in Protégé

VI. Conclusions and Future work

We have proposed a method to exchange a standard CAD format as DXF into OWL and implement it using the API of Protégé 3.4.4. Following the process describe here, it is possible to get an OWL file from another CAD standards as IGES or STEP, and as a future work we will develop these API's and implement them in a software tool to propose OWL as a CAD data exchanger. These API's will be integrated in an architecture as indicated in Fig 6. Each CAD standard will need two API's, one to exchange from the respective standard to OWL (preprocessor) and another to exchange from OWL to the respective standard

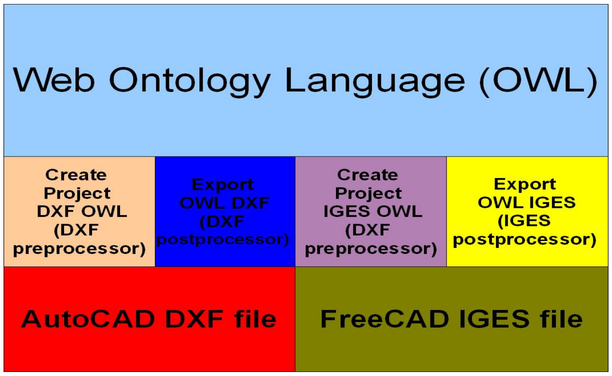

Web Ontology Language (OWL)
Create Project DXF OWL (DXF preprocessor)
Export OWL DXF (DXF postprocessor)
Create Project IGES OWL (DXF preprocessor)
Export OWL IGES (IGES postprocessor)
AutoCAD DXF file
FreeCAD IGES file

Figure 6. Data Exchanger Architecture

(postprocessor). Other elements of the semantic web as the Semantic Web Rule Language will be included in an improved architecture in order to make complete features extraction and generate self-content designs, which could have intelligent interaction in high automated production processes.

ACKNOWLEDGMENT

*This author wants to thanks to the German Foundation for Academical Interchange (DAAD) and to the Fundación Gran Mariscal de Ayacucho (Fundayacucho) who have granted him his PhD scholarship.

REFERENCES

[1] AutoDESK, *AutoCAD 2010*, On line: http://usa.autodesk.com/adsk/servlet/pc/index?id=13779270& siteID=123112, Last access: July, 2010.

[2] TriStar, *ProEngineer*, On line: http://www.tristar.com/software/proengineer.asp, Last acess: July, 2010.

[3] R. Juergen, *Free CAD*, On line: http://sourceforge.net/projects/free-cad/, Last access: July, 2010.

[4] E, Nasr and A, Kamrani. *Computer Based Design and Manufacturing*. Springer Verlag. New York. 2007.

[5] B. Babic, N. Nesic, and Z. Miljkovic, *A review of automated feature recognition with rule-based pattern recognition*, Computers in Industry, 59(4):321-337,2008.

[6] A. Rappoport, *An architecture for universal CAD data exchange*. In Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications (Seattle, Washington, USA, June 16 - 20, 2003). SM '03. ACM, New York, NY, Pages 266-269, 2003.

[7] I. Cayiroglu, *A new method for machining feature extracting of objects using 2D technical drawings*. Computer Aided Design, 41(12):1008 - 1019, 2009.

[8] M. Henderson and D. Anderson, *Computer recognition and extraction of form features: A CAD/CAM link*, Computers in Industry, 5(4):329-339,1984.

[9] A. de Sam Lazaro, D. Engquist, and D. Edwards, *An Intelligent Design for Manufacturability System for Sheet-metal Parts*, Concurrent Engineering, 1(2):117-123,1993.

[10] National Institute of Standard and Technology. *Initial Graphics Exchange Specification IGES 5.3*, On line: http://www.uspro.org/documents/IGES5-3_forDownload.pdf. Last access: July, 2010.

[11] A. Soman, S. Padhye, and M. I. Campbell, *Toward an automated approach to the design of sheet metal components*. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 17(3):187-204, 2003.

[12] A. Ghafour, P. Ghodous, B. Shariat, and E. Perna, *An Ontology-based Approach for Procedural CAD Models Data Exchange*. In Proceeding of the 2006 Conference on Leading the Web in Concurrent Engineering: Next Generation Concurrent Engineering P. Ghodous, R. Dieng-Kuntz, and G. Loureiro, Eds. Frontiers in Artificial Intelligence and Applications, vol. 143. IOS Press, Amsterdam, The Netherlands, 251-259, 2006.

[13] A. Odd and G. Vasilakis, *Building an Ontology of CAD Model Information*. Geometric Modeling, Numerical Simulation, and Optimization Norway: SINTEF, Pages 11-41, 2007.

[14] International Organization for Standarization. *Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles*, On line: http://www.iso.org/iso/catalogue_detail?csnumber=18348. Last access: July, 2010.

[15] M. Grüninger, and A. Delaval, *A First-Order Cutting Process Ontology for Sheet Metal Parts*. In Proceeding of the 2009 Conference on Formal ontologies Meet industry R. Ferrario and A. Oltramari, Eds. Frontiers in Artificial Intelligence and Applications, vol. 198. IOS Press, Amsterdam, The Netherlands, 22-33, 2009.

[16] AutoDesk. *DXF Reference*. On line : http://images.autodesk.com/adsk/files/acad_dxf.pdf. Last access: July, 2010.

[17] C. Guk-Heon, M. Duhwan, and H. Soonhung, *Exchange of CAD part models based on the macro-parametric approach*, in INTERNATIONAL JOURNAL OF CAD - CAM. 2(1): 13-21, 2002.

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

[19] J. Cheng and K. Law, *Using process specification language for project information exchange.* On line: http://eil.stanford.edu/publications/jim_cheng/psl berkeley.pdf. Last access: July, 2010.

[20] R. Schevers and H. Drogemuller, *Converting the industry foundation classes to the web ontology language*. In CSIRO Manuf. and VIC; Inf. Technol., Highett, editors, Semantics, Knowledge and Grid, 2005. SKG '05. First International Conference on, Beijing, Pages 73-83, 2005.

Ontology-based Data Exchange and Integration: an Experience in CyberInfrastructure of Sensor Network Based Monitoring System

Chen-Chieh Feng, Liang Yu
Department of Geography
National University of Singapore
1 Arts Link, Singapore 117570
{geofcc, geoly}@nus.edu.sg

***Abstract* — Scientific research has become interdisciplinary and collaborative, of which sharing and utilizing data in an efficient manner is critical. Data collected for environmental monitoring and modeling, however, often lack semantic information vital for efficient data sharing, thereby causing semantic gaps between the data collection and utilization. The problem is especially acute when data have to be processed without human intervention. To support efficient data sharing, this paper proposes an ontology-based architecture to integrate heterogeneous data. With the help of ontology reasoning, it provides a simpler and more intelligent way for data searching with high-precision and high-recall.**

Keywords - Sensor Network; Ontology Reasoning; Alignment

I. INTRODUCTION

Scientific data are collected and exchanged across many research groups. As the volume of data and range of applications increase, being able to access the data that suit our needs has become a demanding process. Although we could easily access any existing data resources through the Internet, it is often difficult to utilize them due to various heterogeneities between different data sources. Semantic heterogeneity, in particular, presents a major problem for data integration in any interdisciplinary projects [1, 2]. Such problem exists because researchers from different disciplines commit to different domain knowledge and vocabularies, thereby generating semantic gaps that must be bridged before data from different research groups can be integrated.

To cope with this problem, an information system needs to be able to parse and analyze intelligently the search issued by researchers – it should understand user queries, automatically identify data with compatible semantics, and return the data to the researchers. Over years, various tools and technologies have been explored to achieve this goal. Metadata systems have been used to assist determining the usability of datasets [3]. Such metadata systems often work for single domains. Their capability to support data queries in an interdisciplinary project is therefore not guaranteed. Ontology technology, on the other hand, has been introduced to make explicit data semantics and to support data integration at the semantic level. Ontology-driven infrastructure has increasingly gained recognitions [1, 4, 5], especially in spatial science [6, 7]. Several reusable upper ontologies have been developed [8-10]. They provide foundation for developing domain ontologies that can be easily integrated. All these developments suggest that ontology technology is a promising tool to bridge the semantic gaps facing an interdisciplinary project.

The paper aims to develop an ontology-based infrastructure to support semantic data access in an interdisciplinary project involving pervasive monitoring and modeling of the physical environment using sensor-network. Ontology is an explicit specification of conceptualization that encodes inter-connected concepts [11]. It can be extended easily to accommodate an unlimited number of concepts compared to traditional metadata systems. Its capability to support reasoning is extremely advantageous for bridging the semantic gaps as no additional classification or annotation is needed. We developed an ontology and related functions as the core components of a cyberinfrastructure for the sensor network, aiming at helping users from different domains utilize the sensor data efficiently. It has the following components: (1) a user interface that accepts queries from the users and returns query results to the users, (2) a reasoning engine that supports intelligent search, and (3) validators that verify metadata and data formats. The system is distributed and the data in the system are managed in separate databases, each of which stores data developed or processed by a research group. Each group uses a unique set of concepts and constraints to describe the meanings of its datasets. The concepts used to query the data are domain-specific.

The paper is organized as follows. The next section introduces the related work and our approach. Section III discusses our ontology design. Section IV introduces how ontology reasoning is used to facilitate data integration. Section V demonstrates the ontology alignment technique and how to connect the raw data with our ontology. Section VI presents the current system implementation. Section VII provides conclusion and presents future work.

II. RELATED WORK AND OUR APPROACH

Over years ontologies that can potentially be used to support data search and integration have been developed. Upper ontologies such as BFO [10], DOLCE [8], and SUMO [9], were developed as foundation ontologies on which various domain ontologies can be developed and then be used to facilitate data search and integration. SWEET (Semantic Web for Earth and Environmental Terminology) [12], a comprehensive ontology for the earth science domain, has been used in various scientific projects [13, 14]. Its main concepts such as *Data*, *PhysicalProperty*, *Substance*, are critical for describing data semantics. However, it has few relations between concepts, which are essential for reasoning between concepts [15].

The use of ontology to search and integrate data has greatly improved the search result [1]. For example, Couchot [4] used a minimum set of concepts, which he called it reduced ontology, to build up descriptive graphs to summarize the content of the web resources. With fewer constraints than a classical ontology, the reduced ontology is more flexible and easy to use. Shah et al. [5] used ontology to annotate biomedical databases so that the data in the databases can be located with ontology concepts.

Many ontology-based methods have been proposed for data integration within distributed data infrastructures. Beran and Piasecki [16] presented a ontology-driven design for an integrated water data system based on SWEET and GCMD (NASA's Global Change Master Directory). To improve both the recall and precision for data searching in different granularity, they proposed a four-layer ontology: navigation, compound, core and detail, each represents a different abstraction level. The navigation layer contains higher-level concepts that make it easy to visualize the ontology. The compound and core layers contain concepts for assisting users' input. The detail layer contains finer concepts of those in the core layer. These concepts are used during search and for clustering the search result.

Ludäscher et al. [17] proposed a multiple-tier mediation framework for integrating data from different types of data formats, such as database and XML file. The framework aims to alleviate data users from coping with various data formats. They introduced a conceptual model wrapper layer (GM-Wrapper) that encapsulates the methods to access data directly and a generic conceptual model (GCM) layer to which the data access methods are mapped. The GCM is then mapped to an integrated view that provides easy data access for the users.

Based on the degree of efficiency and flexibility, Wache et al. [18] classified ontology-driven data integration approaches into single ontology, multiple ontology, and hybrid ontology approaches. The single ontology approach is efficient; the multiple ontology approach is flexible; the hybrid ontology approach achieves both and is thus the preferred method. Buccella et al. [19] evaluated several well-known geographic data integration systems. The result of their work suggests that most such systems are now ontology-based, but the level the geographic information represented, the degree formal representation of ontologies adopted, and the criteria used to determine how integration should proceed vary from one system to another. They thus recommend full inclusion of geographic information into the integration process, a wider adoption of formal model for ontology representation, and a better assimilation of the geographic knowledge (e.g., quantitative and qualitative relations and scale) in the integration process.

Many annotation schemes have been proposed to tag meaning to the data generated by sensors and to improve the efficiency of data exchange. Russomanno et al. [20] developed OntoSensor, a sensor ontology based on SUMO, SensorML [21] and ISO 19115, to define schema required for geographic information and services. It provided a solid conceptual foundation for sensor itself, but lacks certain concepts related to data processing, e.g., calibration, unit, process chain, and input and output. To bridge the semantic gap between sensor data and to solve the disagreement on methods for data access and exchange, Shankar et al. [22] compared the difference between the adoption of a bottom-up, entity-oriented schema construction approach and a top-down, ontology-based approach in creating a conceptual schema for integrating data generated in a wide area sensor network. They argued that the top-down approach provides semantic commonality and enables better implementation interoperability if adhering to an advertised vocabulary, thereby a higher level of semantic interoperability, is the priority for the system design.

The review shows that various ontology designs for facilitating data sharing have been examined and evaluated. To complete our system, however, more work is needed. To be more specific, we need to perform the following four tasks:

Task 1. Generate requests. A user specifies the query criteria, which includes the theme (e.g., rainfall and temperature) and the constraints (e.g., year, spatial domain, and value ranges). The user does not know the availability of the data that are related to these data types and how they are specified in the system.

Task 2. Parse and analyze requests. The reasoning engine translates the user query to an ontology query using the semantic rules defined in ontology.

Task 3. Retrieve Data. This is the process in which the computer system locates and queries heterogeneous data sources, identify suitable data, and then integrates these data into a usable format. The process relies on the alignment between ontology and data schema.

Task 4. Data production and publishing. Data are original generated by sensors and then processed and reorganized. A dataset needs to be registered and aligned to ontology before it becomes searchable and amenable to integration.

These tasks reflect the need of mediating communication between data provider's and user's sides. Task 1 is different from a traditional concept searching for that the query criteria needs to be made, which concerns the data model and numeric representation rather than a usual domain semantics. Task 2 requires a process to convert the query with the help of ontology reasoning. Task 3 requires the alignment between ontology and different types of data sources. Task 4 requires the semantics to be transferred from the data providers to cyberinfrastructure, for which we need to use the existing metadata to populate the semantics defined in our ontology.

To accomplish these tasks we developed our ontology based on SWEET and complimented it with terms from CSDGM (Content Standard for Digital Geospatial Metadata) [23] and SensorML [21] as they are standards for describing semantics of spatial datasets and sensor systems, respectively. We also explored how our ontology can be used with existing metadata systems, which provides valuable information to populate its concepts and then be used for searching and reasoning.

III. Ontology Design

The domains dealt with in this paper mainly include the monitoring and modeling of urban airshed and ocean water quality. For the first domain, attention has been paid to measure various attributes of air (e.g., temperature and humidity). The characteristics of the buildings that constitute urban canyons, specifically their facades, shapes, and functions, as well as the gaps or holes, such as roads and green spaces in between buildings, are also important concepts for describing the micro-climatic behavior in urban areas. Sensors deployed for measuring these air and building characteristics are stationary.

For the second domain, water quality indicators (e.g., pH) and water characteristics (e.g., temperature and current speed) are the most important concepts. However, significant attention has been paid to the navigational concepts as the readings of these water quality indicators are often taken by sensors mounted on autonomous vehicles. Location information for the individual vehicles or groups of vehicles as well as the location for the potential danger zones (e.g., zones with underwater barriers) are thus important for researchers to make sense of the data collected.

To capture these domain concepts and to support intelligent search across domains, the ontology design adopts a strategy that is in line with the recommendations from the ontology development community – that the ontology is modular [24], has a clearly delineated content [25], is based on a well-designed upper-level ontology [26], and is independent from any databases [11]. The strategy leads to a two-layer ontology that consists of two domain ontologies at the bottom, and cyberinfrastructure (CI) ontology at the top. CI ontology acts as the basis of domain ontology. Concepts in the CI ontology such as *Space* and *Time* are generic to both domain ontologies and are useful for defining domain concepts in a more consistent manner.

Some of the concepts in the SWEET Ontology are adopted in the CI ontology as it provides a common semantic framework for earth science domains. The SWEET concepts such as *NumericalEntity*, *PhysicalProperty*, *Instrument*, *HumanActivity*, and *Unit* were chosen to be the core CI concepts of the following seven CI main categories (Figure 1) due to their relevancy to the domains in question (note that every category includes the related concepts as well as the core concepts indicated by its name) :

1. **Data**. Data is the core concept of the CI ontology and has a much richer meaning than most other concepts because it can be instances of any others. It is also connected to many other concepts, including data accessing forms (e.g., *DataFile* and *DataService*), data format (e.g., *Text* and *Binary*), data attribute (e.g., *Size* and *Format*), and spatial data model (e.g., *Vector* and *Raster*). It is mainly related to the Data in SWEET and the concepts and relations from CSDGM.

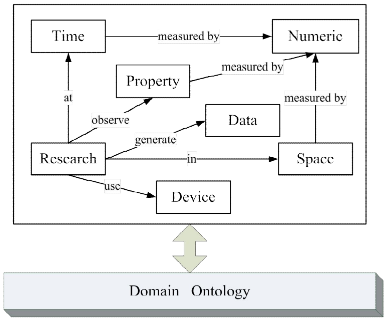


Figure 1. The Main Concepts of CI Ontology

2. **Property**. The concept describes physical and spatial quality associated with an object. For example, physical properties such as *Temperature*, *Weight*, and *Length*, are applicable for most physical objects. Spatial properties such as *Location*, *Orientation*, and *Elevation*, are applicable for objects which are in a space coordinate system. It is mainly related to *PhysicalProperty* of SWEET.
3. **Device.** This concept describes all the hardware used in the research. The most typical ones are *Computer*, *Sensor*, *Vehicle*, and *GPS*. Most of these concepts are sub-concepts of *sweet*:*Instrument*. Since the *sciInstrument* of SWEET has limited sub-concepts, concepts from SensorML and our research domains have been added (e.g., environment and geography).
4. **Research.** This concept incorporate any research domains, research actions (e.g., *Observation* and *Analysis Fieldwork*), and academic activities (e.g., *Conference* and *Publication*). It is mainly related to the *HumanActivity* of SWEET.
5. **Space.** This concept describes the basic characteristics of physical spaces of an object. An object can be associated with one or more two- or three-dimension properties that indicate its geometric characteristics such as location and shape. It also defines the basic frames and reference for spatial objects, including topology relations such as *containment* and *located-in*. *Space* is the basic category of SWEET, and its related *spaceCoordinates*, *spaceDirection*, *spaceDistribution*, and *spaceObject*.
6. **Time.** Temporal concepts are most common in any environmental data. Similar to spatial reference system, time is always associated with a reference system, such as Before Christ (B.C.) and Anno Domini (A.D.). Some computer systems may use other reference systems or their own customized systems. These concepts are related to the *Time* concept in SWEET. There are different units and reference systems for time, which are defined in ontology to assist the processing of temporal data.
7. **Numeric.** Numeric concepts are used to represent the quantity observed for a property. They are associated with values, units and reference systems, which are defined here and reused for specific subclasses. It is mainly related to *NumericEntity* and *Unit* of SWEET.

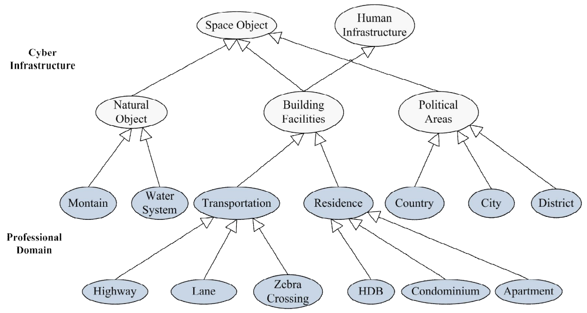


Figure 2. Concepts from different layers of ontology. HDB (Housing Development Bureau Flat, flats built by Housing Development Bureau of Singapore)

The concepts extracted from the domain are included in the domain ontologies. Each of these concepts holds an *is-a* relationship to one or more concepts in the CI ontology. For example, the *Residence* in the domain ontology is a *BuildingFacility* in the CI ontology (Figure 2). Concepts from SWEET are incorporated in domain ontology if they are deemed equivalent to the domain concepts, e.g., *Temperature*, *Humidity*, and *WaterPressure*.

The domain ontologies are enriched with the concepts from CSDGM and SensorML. The CSDGM defines the necessary metadata for a dataset, some of which have relations to *Data*, such as name, URI, spatial domain, spatial reference, size, and suffix, while others are sub-concepts, such as *ShapeFile* as a sub-concept of *DataFile*. SensorML provides a list of sensor concepts for us such as *Thermometer*, *Anemometer*, and *Barometer*. These concepts are all sub-concepts of *sweet:Instrument*. Furthermore, we specify the concepts such as *Input* and *Output*, which could be referred to *PhysicalProperty*, such as *WindPressure*, *WaterTemperature*.

From the development of the two-layered ontology several points were learned. First, existing metadata standards and upper-level ontologies such as SWEET generally contain concepts sufficient for describing the semantics of the environmental data. What is needed, however, is a clear distinction between concepts, relations, and a more comprehensive encoding of the relationships between these concepts. They enable the system to automatically identify and match the related concepts.

Second, the layered ontology is flexible for queries with different granularities. A user of the system can readily query related data and refine the query conditions with the help of the ontology. For example, to identify the temperature data, a user of the system might start with a search based on the concept *Temperature*, and then filter the result by its relations, e.g., spatial and time scope. The user might want to know the usability of this data, which might be met by giving them the sensor information from which the data are generated. Well-formed upper-level concepts and relations between them can be utilized to query the data by both domain and computational concepts.

Third, for the data engineers who are responsible to help both the data providers to publish their data and the data users to find the right data, the need to bridge the gap between domain concepts (e.g., temperature) and computational concepts (e.g., data service, file repository and database) cannot be overlooked. The task requires ontology to recognize computational concepts but not mix them up with the professional domain concepts. Separating the domain concepts in the domain ontology from computational concepts in the CI ontology helps maintain such conceptual clarity. In addition, it utilizes concepts in different ways, alleviating data engineers or providers from doing the conversion between the user interfaces and different cyber components.

IV. REASONING

Reasoning enables multiple interpretations of one or more basic concepts [27]. It also reduces the number of concepts that are left undefined while making precise the semantics of other concepts. In our work, reasoning is supported by three types of information: (1) explicitly declared relations between concepts, (2) T-Box axioms, and (3) rules. The explicitly declared relations, such as is-a, permits the reasoning of related concepts based on the axioms defined with the relations. T-Box axioms can be necessary or equivalent axioms that are used to infer new relations for existing data. The axioms can also be used to infer concepts associated with the concept in question and the relations between them. Such inference mechanism enables validation of the completeness of the data. Rules are specified by the ontology designer to indicate the implications between two sets of statements.

Examples of T-Box axioms are shown in Table 1. Axiom 1 validates if a metadata has provided the basic provenance information, which comes from either observation or process. Axiom 2 validates if a vector instance (e.g., time point) has been assigned reference information, e.g., UTC to a temporal value. Axiom 3 validates the completeness of a process definition. Axiom 4, 5, and 6 populate new concepts using instance from other concepts that make more sense to users from different professional domains.

Table 1. Examples of T-Box axioms**.** The "*some*" means there is at least one value coming from the range defined thereafter. The "*min*", "*max*" and "*exactly*" are cardinality constraints on the binary relations.

	T-Box Axioms
1.	Data ⊂ {has_source *some* Observation ∪has_source *some* Process}
2.	Vector ⊂ {Numeric ∩ has_reference *exactly* 1 }
3.	Process ⊂ {has_input min 1 ∩ has_output *min* 1 ∩ has_processor *min* 1}
4.	GeographicalData ≡ {Data ∩ has_model *some* SpatialDataModel}
5.	ElevationData ≡ {Data ∩ (has_model *some* DEM ∪has_model *some* DTM ∪has_model some DSM ∪has_model *some* Contour)}
6.	Thermometer ≡ {Sensor ∩ has_input *some* Temperature }

Table 2. Rules for the reasoning on data. The “r(x,y)” means that binary relation r has the subject x and the object y. The “ist(x,y)” means that x is an instance of y. The “sub(x,y)” means x is a subclass of y. The “sup(x,y)” means x is a super class of y. The “eql(x, y)” means x equals to y.

	Rule
1.	has_content(?x,?c1)∩has_content (?y,?c2)∩(sub(?c1,?c2)∪ sup(?c1,?c2)∪eql(?c1,?c2)) → compatible_content(?x,?y)
2.	ist(?p,Process)∩has_input(?p,?x)∩has_output(?p,?y)→ has_parent(?x,?y)
3.	ist(?x,TemperatureUnit)∩ist(?y,TemperatureUnit)→ convertible_unit(?x,?y)
4.	ist(GeoReference,?x)∩ist(GeoReference,?y)→ convertible_reference(?x,?y)

5.	ist(?x,Contour)∩ist(?y,DEM)→convertible_model(?x,?y)
	ist(?x,DEM)∩ist(?y,TIN)→convertible_model(?x,?y)
	ist(?x,DLG)∩ist(?y,DLG)∩has_feature_type(?x,?f)∩ has_feature_type(?y,?f)→convertible_model(?x,?y)

6.	ist(?x, Data) ∩generatedBy(?x, ?s) ∩has_location(?s, ?p) → located_in(?x, ?p)

A portion of the rules which had been useful in supporting reasoning on the datasets and sensors in our work is shown in Table 2. Rule 1 is used to decide if two datasets contains the same domain concepts, which indicates the compatibility of the datasets. Rule 2 is used to infer the provenance relation between two data sets. Rule 3 indicates that units under the same category are compatible and amenable to conversion, which is useful to deciding if two numeric instances with the specified units are convertible. Rule 4 indicates whether geo-reference systems are convertible to each other, e.g., a local coordinate system without geo-reference components such as datum, projection, is not convertible to a geo-reference system. Rule 5 indicates four pairs of model which are considered as compatible. Rule 6 makes the data generated by the sensor inherit some relations from it. In this case, the location of the sensor is taken as the location of the data.

All data in our project can be the input of the reasoning engine. Figure 3 shows the correspondence between the CSDGM metadata and ontology concepts. The reasoning engine does the conversion by parsing the CSDGM metadata entries and creating instances of its corresponding ontology concepts, e.g., instances of *DataFile*. The reasoning engine also performs validation and inference during conversion by using all axioms and rules, which include the declared “is-a” relations and those from the above two tables.

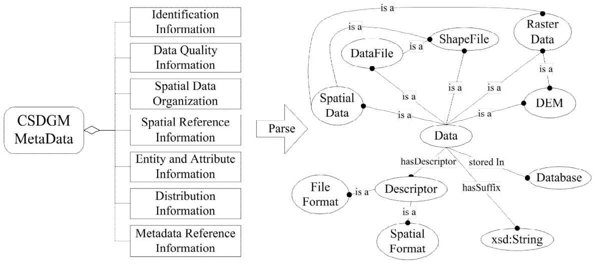


Figure 3. Parse the CSDGM metadata to ontology structure.

In our work, semantic gap exists between users of different domains. Reasoning function can be used to search the result with high precision and recall with the criteria which is not specified in the original query but stated either explicitly or implicitly in the ontology. Moreover, for the data engineers, reasoning function helps them to efficiently develop the program. Since axioms can be updated on the fly, costs of updating the programs in an ever-changing project can be sharply reduced.

V. Alignment and translation

Ontology alignment is the process of establishing correspondence between two similar concepts, including their subordinate and related concepts. In a data-centered scientific research, an alignment mechanism is needed to extract information from the data models of the original data sources, to perform reasoning, and to translate an ontology query request to a specific query language. A tool to support such process is vital in our work because the data producers and users often use different terms to refer to the same concepts and different encoding methods for their data.

Three types of alignment between commonly used data models and our domain ontologies were explored. Using the alignment between the CSDGM metadata and our domain ontologies (Figure 4), they include:

1. **Concept alignment** that identifies the corresponding concepts by text comparison. For example, the keyword *Depth* in an AUV dataset is identified as *Depth* (*of Water*). This alignment process is usually facilitated by referring to the context, i.e., the standard vocabulary or the ontology, used by the users.
2. **Instance alignment** that identifies the correspondence between instances in an ontology concept and semantic information in the database. These instances are typically extracted from rows in a database table or identified by unique reference identifiers.
3. **Relation alignment** that identifies the attribute of a data model to be a relation of an ontology concept, such as a temperature value of a “Temperature” concept, and other concepts essential to define the attribute, such as the unit for temperature.

One benefit of establishing these alignments is to facilitate the conversion of heterogeneous data models into a global ontological model preferred by the users of a particular domain [28]. Consider the following query: retrieve the climate data of Asia and return the records which were produced between *2009-3-3* and *2009-4-3*. To accomplish this task, the first step is to infer the possible candidates which meet the semantic requirement. In this case, there are two semantic restrictions: *is-a Climate* and *located-in Asia*. The relations *is-a* and *located-in* are both transitive relations. The reasoning process returns a list of candidate datasets related to *Climate*, such as *Temperature*, *Humidity*, and are located in *Asia*, e.g., *Japan* and *Singapore*. To further narrow in on the data in a particular time frame, assuming the data are stored in relational tables, the query with a time filter will be sent to the mediator for the relational tables and translated to the following:

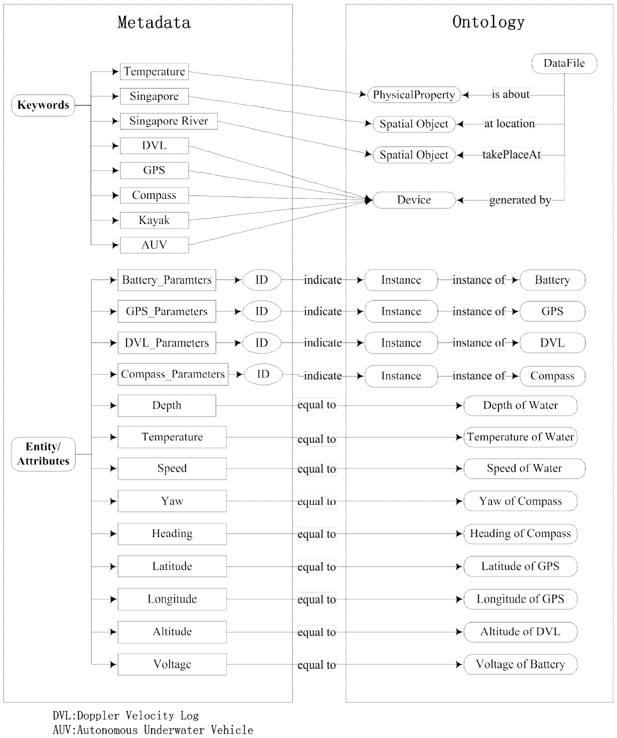


Figure 4. An example of the alignment between the metadata and the ontology.

Select value1, time1 from table1 where time1 > ‘2009-3-3’ and time1 <’2009-4-3’

Select value2, time2 from table2 where time2 > ‘2009-3-3’ and time2 <’2009-4-3’

Select value3, time3 from table3 where time3 > 1236009600000 and time3 <1238688000000

The last clause uses integer values for time representation (milliseconds between 1970-1-1 00:00:00 GMT and this time), which is used by some systems and needs to be translated to a uniform format. Alignment can take advantage of the *is-a* relation encoded in ontology concepts, i.e., if an alignment is applicable to an ontology concept, it could be applied to all its sub-concepts as well.

A second benefit of establishing these alignments is to support automatic conversion between scalars, between vectors, and between data formats:

1. **Scalar conversion**. This is a conversion for the data values described by a single scalar and an associated unit, such as different unit for Length, Area, Time, and Pressure.
2. **Vector conversion**. This is a conversion between data whose values are referenced to a chosen reference system. Spatial coordinates and time are typical examples of a vector.
3. **Text format conversion**. This is a conversion between different representation formats. For example, the text format for *Date* and *Time* varies in different data, even if they use the same unit and the reference system.

In our system, we focus on the alignment between ontology and conceptual or systematic data model. It is a process similar to ontology alignment except that a common data model is always vague on semantics, i.e., different types of entities and relations between them, where entity here can be regarded as an instance of a specified concept. Thus, the first step is to rebuild the entities and relations of the data model. In our system, we utilize ORM (Object-Relational Mapping) tool to achieve this on relational databases.

VI. IMPLEMENTATION

The system is a web application based on J2EE. We use Java as the major implementation language because it is widely supported by the open sources communities. We selected two ontology projects – Jena [28] and Pellet [29] – to process the ontology files. The OWL files are firstly generated by Protégé, and then stored and maintained via Jena API. The Pellet is a reasoning engine that provides support for SWRL (Semantic Web Rule Language [30]) based rules. We use ArcGIS from ESRI to develop spatial-related functions. The system architecture is shown in Figure 5.

The system adopts a three-layer architecture – the UI and Application Layer, Data Registration Layer, and the Resource Layer. The Data Registration layer uses the ontologies and other APIs to process the original information, which contains two paths, one for the data users and the other for the data providers:

For a data user, it is easier and more straightforward to search by concepts rather than look into the details of data. The user therefore uses the ontology to develop a request, which will be processed by the reasoning engine in the system and attached with richer semantics (e.g., more concepts and restrictions). The original query will be translated to different forms suitable for querying different data sources with the ontology alignment service.

A data provider provides sufficient metadata based on uniform standards, such as an XML schema which can be used to standardize the format of metadata, and is useful for standard-dependent programs. The metadata could be directly referred to ontology concepts, or some other standard vocabulary like GCMD, whereby they would then be recognized and aligned automatically. The data provider can also add more alignments or alter existing ones manually. Both of the original metadata and generated alignment are stored in the database for future applications.

VII. CONCLUSION AND FUTURE WORK

We have used ontology to model data semantics and to help users unfamiliar with the data structure and semantics to find the data they look for. We demonstrated the advantage of ontology-based system over traditional metadata or standard data exchange systems in bridging the semantic gaps in a heterogeneous environment. Ontology reasoning is a powerful tool for generating or importing a new ontology and its concepts without modifying the data. It makes it possible to accommodate concepts across different domains and from different user groups. Ontology alignment acts as a middleware for integrating data from different sources.

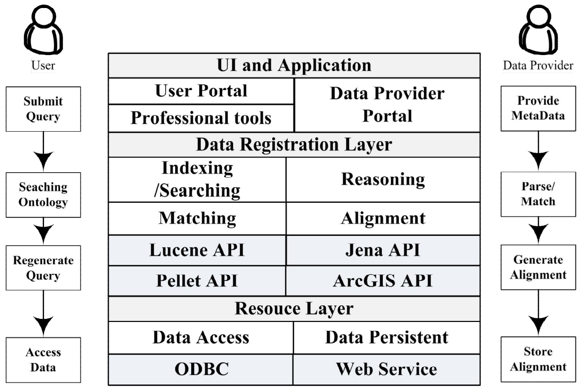


Figure 5. System architecture.

Compared to the most existing methods such as those mentioned in Section II, our work has focused on the following two points:

1. We have considered how to support data integration at both the semantic level and the conceptual model level. We give a clear roadmap from the users' request to the data retrieval, along with which ontology reasoning is essential for dataset searching and for integration.
2. We have designed and implemented a modular data integration system to ensure system flexibility. Every single application works independently while cooperating with each other through a dynamic, semantics-enabled interface. We also make sure the system is connected to the existing technologies and systems so that existing tools, e.g., metadata for populating databases and alignments for translating user queries to database queries, are reused.

Interfaces useful for automating data registration and alignment were developed. They allow a data set to be registered by uploading the associated metadata file compatible with CSDGM. They then automatically create instances of the metadata entries using the ontologies in the system. Alignments are mainly performed by the system managers who are well versed with the ontology concepts. Through text matching and ontology reasoning, the interfaces suggest the necessary inputs associated with the data that will be uploaded to the system.

Yet more remain to be incorporated to enrich the functionality of this system. First, with the increase of new applications and users, ontology is bound to evolve through time [31]. How to ensure the consistency of the whole ontology while evolving is an important problem to investigate. Second, extending the spatial reasoning capability of the system is crucial. For example, we can use spatial computation in the reasoning process to define the relations as *near*, *far*, and *neighbor*. Integrating spatial functions with ontology components particularly the reasoning functions would significantly improve the data search capability of the system. Third, users should be able to share both the data as well as the services associated with the formats of the data. For example, users could click a link to view a searched spatial dataset via an online visualization service. In this process, data should be organized and converted to a specific format suitable as the input of the service. The users can also choose to download them in a specific format and use local tools to handle them.

REFERENCES

[1] O. Dridi, “Ontology-Based Information Retrieval: Overview and New Proposition,” Proc. 2nd International Conference on Research Challenges in Information Science (RCIS 2008), IEEE Press, Jun. 2008, pp. 421-426, doi: 10.1109/RCIS.2008.4632133.

[2] B. Barkallah and S. Moalla, "Metadata Driven Integration Model for Large Scale Data Integration," Proc. 7th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 09), IEEE Press, May. 2009, pp. 41-46, doi: 10.1109/AICCSA.2009.5069296.

[3] C. Youn, K. Kaiser, C. Santini and D. Seber, “Developing Metadata Services for Grid Enabling Scientific Applications,” Proc. 6th International Conference on Computational Science (ICCS 2006), Springer-Verlag, May. 2006, pp. 379-386, doi: 10.1007/11758501_53

[4] A. Couchot, “Improving Web Searching Using Descriptive Graphs,” Proc. Natural Language Processing and Information Systems (NLDB 2004), Springer-Verlag, Jun. 2004, pp. 276-287, doi: 10.1007/978-3-540-27779-8_24.

[5] N. H. Shah, C. Jonquet, A. P. Chiang, A. J. Butte, R. Chen and M. A. Musen, “Ontology-driven indexing of public datasets for translational bioinformatics,” BMC Bioinformatics, vol. 10, Feb. 2009, doi: 10.1186/1471-2105-10-S2-S1.

[6] F. Fonseca and M. A. Rodriguez, “From geo-pragmatics to derivation ontologies: New directions for the geospatial semantic web,” Trans. GIS, vol. 11, 2007, pp. 313-316.

[7] J. S. Madin, S. Bowers, M. P. Schildhauer and M.B. Jones, “Advancing ecological research with ontologies,” Trends Ecol. Evol., vol. 23, 2008, pp. 159-168.

[8] 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, vol. 2473, Springer-Verlag, Oct. 2002, pp. 223-233, doi: 10.1007/3-540-45810-7_18.

[9] “Suggested Upper Merged Ontology (SUMO),” http://www.ontologyportal.org/, [accessed 17 December].

[10] “Basic Formal Ontology (BFO),” http://www.ifomis.org/bfo, [accessed 16 July 2010].

[11] T. R. Gruber, “A Translation Approach to Portable Ontology Specifications,” Knowl. Acquis., vol. 5, 1993, pp. 199-220.

[12] “Semantic Web for Earth and Environmental Terminology (SWEET),” http://www.jpl.nasa.gov/ontology/, [accessed 17 July 2010].

[13] R. G. Raskin and M. J. Pan, “Knowledge representation in the semantic web for earth and environmental terminology (SWEET),” Comput. Geosci., vol. 31, Nov. 2005, pp. 1119-1125.

[14] A. Tripathi and H. A. Babaie, “Developing a modular hydrogeology ontology by extending the SWEET upper-level ontologies,” Comput. Geosci., vol. 34, Sep. 2008, pp. 1022-1033.

[15] B. Brodaric and F. Probst, “DOLCE Rocks: Integrating Geoscience Ontologies with DOLCE,” Proc. 2008 AAAI Spring Symposium, American Association for Artificial Intelligence (AAAI 2008), Mar. 2008, pp. 3-8.

[16] B. Beran and M. Piasecki, “Engineering New Paths to Water Data,” Comput. Geosci., vol. 35, no. 4, Apr. 2009, pp. 753-760.

[17] B. Ludäscher, A. Gupta, and M. E. Martone, “Model-based Mediation with Domain Maps,” Proc. 17th IEEE International Conference on Data Engineering (ICDE 2001), IEEE Comput. Soc., Apr. 2001, pp. 81-90, doi: 10.1109/ICDE.2001.914816.

[18] H. Wache, T. Vogele, and U. Visser, “Ontology-Based Integration of Information - A Survey of Existing Approaches,” Proc. 17th International Joint Conferences on Artificial Intelligence (IJCAI 2001), Morgan Kaufmann, 2001, pp. 108-117.

[19] A. Buccella, A. Cechich, and P. Fillottrani, “Ontology-driven geographic information integration: A survey of current approaches,” Comput. Geosci., vol. 35, Apr. 2009, pp. 710-723.

[20] D. J. Russomanno, C. R. Kothari, and O. A. Thomas, “Building a Sensor Ontology: A Practical Approach Leveraging ISO and OGC Models,” Proc. International Conference on Artificial Intelligence (ICAI 2005), CSREA Press, Jun. 2005, pp. 637-643.

[21] “OGC, Sensor Model Language (SensorML),” http://www.opengeospatial.org/standards/sensorml, [accessed 17 July 2010].

[22] M. Shankar, A. Sorokine, B. Bhaduri, D. Resseguie, S. Shekhar, and J. S. Yoo, “Spatio-Temporal Conceptual Schema Development for Wide-Area Sensor Networks,” Geospatial Semantics, vol. 4853, Springer-Verlag, Nov. 2007, pp. 160-176, doi: 10.1007/978-3-540-76876-0_11.

[23] “FGDC, Content Standard for Digital Geospatial Metadata,” http://www.fgdc.gov/metadata/csdgm/, [accessed 17 July 2010].

[24] R. Rector, “Modularization of Domain Ontologies Implemented in Description Logics and Related Formalisms including OWL,” Proc. 2nd International Conference on Knowledge Capture (K-CAP 03), ACM, 2003, pp. 121-128, doi: 10.1145/945645.945664.

[25] “The Open Biological and Biomedical Ontologies: Current Principles,” http://www.obofoundry.org/crit.shtml. [accessed 17 July 2010].

[26] N. Guarino, “Formal Ontology in Information Systems,” Proc. 1st International Conference on Formal Ontology in Information Systems (FOIS 98), IOS Press, 1998, pp. 3-15.

[27] J. Z. Pan and I. Horrocks, “Web Ontology reasoning with Datatype Groups,” Proc. 2nd International Semantic Web Conference (ISWC2003), Springer-Verlag, Oct. 2003, pp. 47-63, doi: 10.1007/978-3-540-39718-2_4.

[28] “Jena - A Semantic Web Framework for Java, ” http://jena.sourceforge.net/, [accessed 17 July 2010].

[29] “Pellet: The Open Source OWL Reasoner, ” http://clarkparsia.com/pellet/, [accessed 17 July 2010].

[30] “OGC, SWRL: A Semantic web Rule Language Combining OWL and RuleML, ” http://www.w3.org/Submission/SWRL/, [accessed 17 July 2010].

[31] G. Flouris, D. Plexousakis, and G. Antoniou, “Evolving Ontology Evolution, ” SOFSEM 2006: Theory and Practice of Computer Science, vol. 3831, Springer, Jan. 2006, pp.14-29, doi: 10.1007/11611257_2.

Document Clustering Using Semantic Relationship Between Target Documents and Related Documents

Minoru Sasaki
Dept. of Computer and Information Sciences
Faculty of Engineering, Ibaraki University
4-12-1, Nakanarusawa, Hitachi, Ibaraki, 316-8511, Japan
Email: msasaki@mx.ibaraki.ac.jp

Hiroyuki Shinnou
Dept. of Computer and Information Sciences
Faculty of Engineering, Ibaraki University
4-12-1, Nakanarusawa, Hitachi, Ibaraki, 316-8511, Japan
Email: shinnou@mx.ibaraki.ac.jp

***Abstract*—Document clustering is one of the most major techniques to group documents automatically. This technique is to divide a given set of documents into a certain number of clusters automatically. In this technique, the first step is 'feature extraction' from documents. As a feature used in the conventional methods, we frequently use a set of words that contains nouns and verbs. Although words are used as features in a generic clustering framework, some previous research proposes the clustering method using the other features based on vector space model such as kernel methods and adaptive sprinkling. However, in previous research of document clustering, the method of appending new feature vectors obtained by using relationship between the existing documents and other documents has not been reported yet. So, we propose a new method for clustering documents using the relationship between the existing documents and other documents to acquire the more useful clusters for users. Our method can expand features of document similarities as semantic relationships by using relevant documents that user is interested in, like semi-supervised clustering. To evaluate the efficiency of this system, we made experiments on clustering newsgroup documents by using our method and by using the dimension reduction method based on the singular value decomposition. As the results of these experiments, we found that (i) it is effective for document clustering to combine the similarity matrix with the original matrix, and (ii) low similarity values cause adverse effect to the clustering performance when we use all the similarity value. Moreover, the proposed method is more effective for the document clustering in comparison with the clustering through the dimensionality reduction.**

***Keywords*-document clustering; semi-supervised clustering; semantic feature expansion;**

I. Introduction

Document clustering is one of the most major techniques to group documents automatically. This technique is to divide a given set of documents into a certain number of clusters automatically. Each cluster obtained by this technique represents a topic, which is different from the other topics. Thus, it enables a user to have an overall view of the topics contained in the documents so that this technique is often applied to the analysis of web data [13], news articles [12], patents and research papers [1] and so on.

In the document clustering, the first step of preprocessing is term extraction from a set of documents. After the term extraction process, various clustering methods can be applied by utilizing these extracted characteristics of terms. As a feature used in the conventional methods, we frequently use a set of words that contains noun and verb words obtained by using a morphological analyzer from the documents. For the set of these terms, the weight of each word in each document is calculated by using term weighting methods such as term frequency (TF), inverse document frequency (IDF) or log-likelihood ratio to construct a term-document matrix.

Although words are used as features in a generic clustering framework, some previous research proposes the clustering method using the other semantic features based on vector space model. For example, co-clustering methods [5] [8], which is the simultaneous clustering of both words and documents, partitions the documents using word cluster as the feature. The kernel trick [6], which is used to measure the similarity (or distance) of vectors, enables the computation of inner product in a space of possibly very high dimension by some linear combination of words as the feature. Moreover, adaptive sprinkling [4] is effective method to obtain feature vectors by appending some principal component vectors to the term-document matrix by using the singular value decomposition (SVD).

As mentioned above, the co-clustering method and the kernel trick method produce new features obtained by using the relationship between existing words. However, in previous research of document clustering, the method of appending new feature vectors obtained by using relationship between the existing documents and other documents has not been reported yet. The similarity between relevant documents increases with additional feature of other documents so that we consider these features to be efficient for users to obtain useful clustering results. For this reason, we propose a new method for clustering documents using the relationship between the target documents and the other documents. To evaluate the efficiency of this system, we make experiments on clustering newsgroup documents by using our method. Moreover, as comparative experiment we make an experiment by using the dimension reductio method based on the singular value decomposition.

II. Related Works

We consider this proposed method to be positioned as on of the semi-supervised clustering [3] [9]. Our objective o the method is to improve the efficiency of clustering results by providing related information. In the standard semi-supervised clustering, it is hard to find similar (or dissimilar) document pairs. However, the proposed method use related data as additional features like must-link constraints so it is easy to provide the constraints by comparison with the semi-supervised clustering. In the co-clustering algorithm [5] [8], features are first grouped to perform document clustering. In contrast, the proposed method uses the features that consist of both the original bag of words and the group of words. The kernel method [6] computes the inner product in a space of possibly very high dimension by some linear combination of words. It is similar to the proposed method in the use of additional features. However, the kernel method is difficult to find efficient combination of words from the high dimensional space. Moreover, adaptive sprinkling method [4] appends some principal component vectors of the term-document matrix to obtain the effective features. In contrast, the proposed method appends new feature vectors obtained by using relationship between the existing documents and other documents.

III. Clustering Method Based on Relationship Feature Expansion

A. Motivation

In previous researches, there are some methods that insert additional features obtained by using the relationship between existing words. For example, there is a method that creates combinations of features using kernel methods, and another method that learns similarity metric by information that consists of a set of similar(dissimilar) pair such as semi-supervised clustering. However, sometimes it is hard to construct the additional information of pairs, even though the semi-supervised methods use only a small number of pairwise relations. For this reason, we propose a new method for clustering documents using the relationship between the existing documents and other documents.

The similarity between relevant documents increases with additional feature of other documents. We show an example to explain the reason of this efficiency. In the Figure 1, we consider that there are four documents A, B, C and D in the target document set. The similarity between the document and the other two is nearly equal (e.g., A and B, A and D) so that it is difficult to cluster these data (in the Figure 1 a). Then, we consider another document X in which a user needs relevant information. The similarities between the X and the target document set are calculated and added to their document vectors as an additional features (in the Figure 1 b) so that the similarity between the target documents changes with the relationship between the target and the additional documents. Therefore, the document A and B are more similar than the C and D and the C and D are more similar than the A and B so that we are able to acquire the more useful clusters for users from the target document set (in the Figure 1 c).

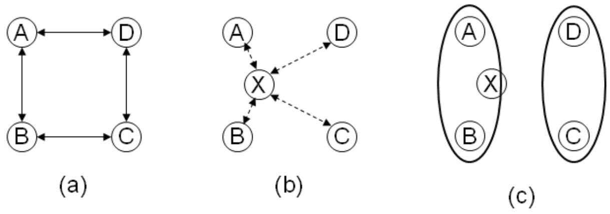


Figure 1. Description of the efficiency by expanding correlation features

Our proposed method resembles the kernel methods in the point of dimensionality expansion. The kernel methods calculate the weights of additional features that are combined by the existing features. Then, some of these additional features work well for learning appropriate document vectors. However, Our method is possible to expand features of document similarities by using relevant documents that user is interested in like semi-supervised clustering.

B. Keywords Extraction

As features of the clustering, we extract words from documents from the 20 Newsgroups data set [11]. We first preprocessed all documents in the documents to remove all the stop words using a stop list of common English words such as “a” or “about”. For the obtained words, we calculate the relevancy of each word with respect to each document by using TF-IDF weighting scheme [14]. Therefore, a term-document matrix is generated by normalizing document vectors as shown in the upper part in the Figure 2 .

C. Method of Feature Expansion

For this term-document matrix, we combine a similarity matrix between target documents and other documents to construct an expanded matrix. As a first step, we provide other relevant documents which are different to the target documents. Next, we extract words from the additional documents in the same way as the word extraction from the target documents. Then, we construct a term-document matrix of the additional documents as described in lower part of the Figure 2, where the terms of this matrix are same as the terms appearing in the target documents. To make the correlation matrix, we calculate document similarity matrix based on the cosine similarity measure. Finally, we combine the term-document matrix of the target documents and this similarity matrix by rows to construct the expanded matrix.

ISBN: 978-1-61208-104-5

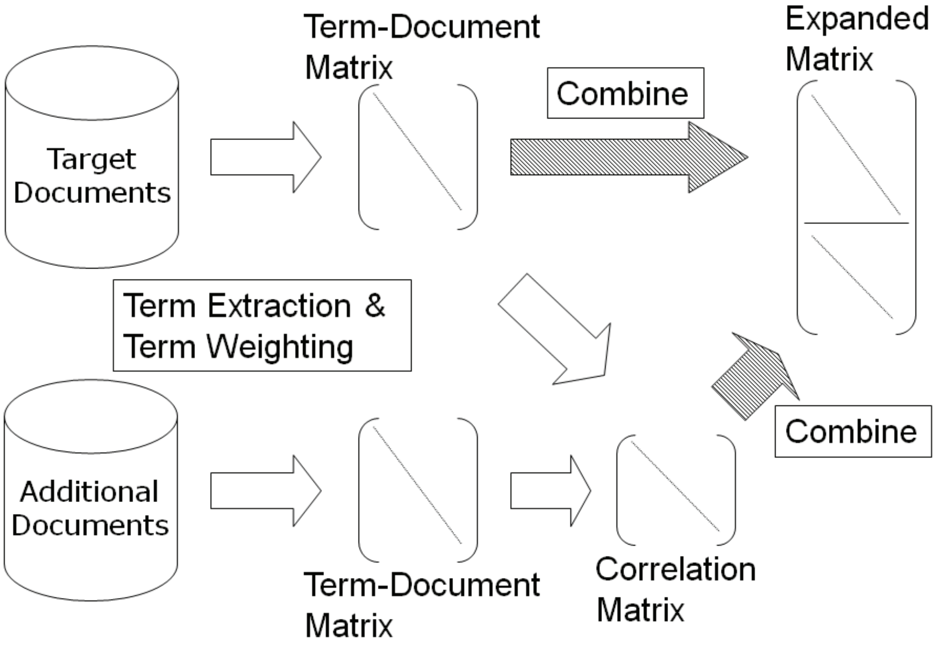


Figure 2. Process of constructing an expanded matrix

However, when we use all values in the similarity matrix, our system tends to have a higher sensitivity to statistical noise due to low values of this matrix. To solve this problem, for all the elements of the document vector, we use the top k similarity values and set all other similarity values to 0. k is defined as the number of the similarity scores. This process enables to reduce the noise in the similarity matrix.

D. Clustering Method

For the expanded matrix generated as mentioned above, we apply a clustering method to group similar documents in the target documents. Let the expanded matrix M, which is obtained above to be the $(m+l) \times n$ matrix, consisting of the number of terms m, the number of additional documents l and the number of the documents n. Then our system groups the documents to generate clusters using the matrix M. In our system, we use CLUTO [10] as the tool for clustering documents. For the purpose of evaluation of the efficiency of document clustering for the expanded matrix, we apply the same clustering method of CLUTO in all our experiments.

IV. EXPERIMENTS

To evaluate the efficiency of the proposed method, we make some experiments on clustering newsgroup documents. In this experiment, we use the 20 newsgroups as a document set. The 20 newsgroups data set is a collection of approximately 20000 articles from 20 Usenet newsgroups. We extract 50 articles from each 10 newsgroups and construct the subset which consists of the total 500 articles. We also extract 100 articles from each 10 newsgroups and construct another subset which consists of the total 1000 articles. As additional documents, we extract 50 articles from each 20 newsgroups and construct the subset which consists of the total 1000 articles. For the similarity matrix, the number of top-ranked similarity values which are used in each document vector is varied from 100 (set 900 values to 0) to 1000 (use all values) incremented by 100. Then, we compare the performance using the proposed method with that using only the target documents as a baseline method for the evaluation of our method.

A. Evaluation Measures

In this paper, we use entropy and purity to evaluate the clustering quality [2]. The purity is defined as the degree to which each cluster contains documents primary from a single class. The purity of a clustering result is obtained as a weighted sum of the purity of individual clusters as follows,

$$Purity = \sum_{i=1}^{C} \frac{1}{N} \times \max_{j}(n_{ij}), \qquad (1)$$

where N is the total number of documents, C is the number of clusters and n_{ij} is the number of documents of the

ISBN: 978-1-61208-104-5

Table I
CLUSTERING RESULTS 1 WITH 500 DOCUMENTS FOR THE EACH NUMBER OF SIMILARITY VALUES

The number of top-ranked similarity values	Entropy	Purity
None	0.400	0.686
1000	0.415	0.648
900	0.415	0.620
800	0.415	0.620
700	0.415	0.620
600	0.406	0.676
500	0.406	0.676
400	0.415	0.620
300	0.415	0.620
200	0.410	0.672
100	0.410	0.672

Table II
CLUSTERING RESULTS 2 WITH 500 DOCUMENTS FOR THE EACH NUMBER OF SIMILARITY VALUES

The number of top-ranked similarity values	Entropy	Purity
None	0.371	0.698
1000	0.402	0.664
900	0.360	0.690
800	0.306	0.768
700	0.306	0.768
600	0.312	0.760
500	0.312	0.760
400	0.312	0.760
300	0.312	0.760
200	0.312	0.760
100	0.306	0.768

Table III
CLUSTERING RESULTS 1 WITH 1000 DOCUMENTS FOR THE EACH NUMBER OF SIMILARITY VALUES

The number of top-ranked similarity values	Entropy	Purity
None	0.379	0.755
1000	0.378	0.696
900	0.327	0.779
800	0.328	0.774
700	0.363	0.762
600	0.328	0.774
500	0.328	0.774
400	0.328	0.774
300	0.328	0.774
200	0.356	0.731
100	0.327	0.779

Table IV
CLUSTERING RESULTS 2 WITH 1000 DOCUMENTS FOR THE EACH NUMBER OF SIMILARITY VALUES

The number of top-ranked similarity values	Entropy	Purity
None	0.315	0.750
1000	0.300	0.764
900	0.308	0.750
800	0.308	0.750
700	0.295	0.767
600	0.284	0.780
500	0.284	0.780
400	0.295	0.767
300	0.322	0.752
200	0.284	0.780
100	0.293	0.777

category j in the cluster C_i. In general, the larger the purity value are obtained, the clustering algorithm is the better.

The entropy of a clustering result is defined as the weighted sum of cluster entropies as follows,

$$Entropy = -\sum_{i=1}^{C} \frac{n_i}{N} \sum_{j=1}^{K} \frac{n_{ij}}{n_i} \log \frac{n_{ij}}{n_i}, \qquad (2)$$

where n_i is the number of documents in the cluster C_i. A good clustering algorithm should have low cluster entropy.

V. EXPERIMENTAL RESULTS

A. Clustering Results with 500 Documents

Table I and Table II show the results of the experiments with two sets of 500 articles, which consist 50 articles from each 10 newsgroups as a mentioned above, respectively. In the Table I, the precision is the approximately same as the result using the original matrix. If the additional documents have little association with the target documents about these contents, the clustering performance is less affected by the similarity matrix. However, in the Table I, the purity score represents a 6.2% increase by the addition of the similarity matrix. When the additional documents are relevant to the original documents, we found that it is effective for document clustering to combine the similarity matrix with the original matrix.

Additionally, the system provides the highest accuracy when we use the top 500-600 similarity values in the Table I and the top 700-800 similarity values in the Table II. We found that low similarity values cause adverse effect to the clustering performance when we use all the similarity value.

B. Clustering Results with 1000 Documents

Table III and Table IV show the results of the experiments with two sets of 1000 articles, which consist 100 articles from each 10 newsgroups as a mentioned above, respectively. Though these results are smaller accuracy than that with the 500 documents, the clustering performance is improved by the addition of similarity matrix. This shows that the addition of the similarity matrix is effective for the clustering performance even when we change the number of documents. When we change the number of top similarity value, we obtain the highest accuracy by using the top 500 values.

C. Comparison with Clustering Through Dimensionality Reduction

To evaluate the efficiency of the proposed method, we make another experiment using the clustering through di-

Table V
CLUSTERING RESULTS FOR EACH REDUCED DIMENSIONS (1000 DOCUMENTS)

dimension	Entropy	Purity
900	0.314	0.730
800	0.309	0.728
700	0.252	0.786
600	0.326	0.693
500	0.264	0.769
400	0.325	0.729
300	0.314	0.730
200	0.310	0.736
100	0.328	0.712
50	0.355	0.675
10	0.389	0.626
5	0.478	0.544

mensionality reduction. We compute the singular value decomposition for the term-document matrix generated from the above 1000 documents [7]. The documents are projected in a lower dimensional space spanned by the leading l left singular vectors to obtain dimension reduced vectors. Then our system groups these vectors to generate clusters by the same clustering algorithm.

Table V shows the results of this experiment with the 1000 documents. In this Table V, the clustering accuracy drops continuously as the number of dimensions grows. The projection transforms a document's vector in n-dimensional word space into a vector in the k-dimensional reduced space. Because the characteristics of words are reduced by this projection, it is difficult to make clear distinction between words.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we proposed a new method for clustering documents using the relationship between the existing documents and other documents. To evaluate the efficiency of this system, we make experiments on clustering newsgroup documents by using our method and by using the dimension reduction method based on the singular value decomposition. As the results of these experiments, we found that it is effective for document clustering to combine the similarity matrix with the original matrix and low similarity values cause adverse effect to the clustering performance when we use all the similarity value. Moreover, the proposed method is more effective for the document clustering in comparison with the clustering through the dimensionality reduction.

Further work would be required to compare the other semi-supervised clustering methods by the many kinds of document data.

REFERENCES

[1] B. Aljaber, N. Stokes, J. Bailey, and J. Pei, "Document clustering of scientific texts using citation contexts," *Information Retrieval*, vol. 13, no. 2, pp. 101–131, 2010.

[2] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information Retrieval*, vol. 12, no. 4, pp. 461–486, 2009.

[3] S. Basu, A. Banerjee, and R. J. Mooney, "Semi-supervised clustering by seeding," in *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 27–34.

[4] S. Chakraborti, R. Mukras, R. Lothian, N. Wiratunga, S. Watt, and D. Harper, "Supervised latent semantic indexing using adaptive sprinkling," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI07)*, 2007, pp. 1582–1587.

[5] H. Cho, I. Dhillon, Y. Guan, and S. Sra, "Minimum sum squared residue co-clustering of gene expression data," in *Proceedings of the Fourth SIAM International Conference on Data Mining*, 2004, pp. 114–125.

[6] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[7] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41(6), pp. 391–407, 1990.

[8] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD03)*. New York, NY, USA: ACM, 2003, pp. 89–98.

[9] N. Grira, M. Crucianu, and N. Boujemaa, "Unsupervised and semi-supervised clustering: a brief survey," in *'A Review of Machine Learning Techniques for Processing Multimedia Content', Report of the MUSCLE European Network of Excellence (FP6)*, 2005.

[10] G. Karypis, *CLUTO - Software for Clustering High-Dimensional Datasets : A Clustering Toolkit, Release 2.1.1*, http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview, 2003.

[11] K. Lang, "Newsweeder: Learning to filter netnews," in *Proceedings of the Twelfth International Conference on Machine Learning*, 1995, pp. 331–339.

[12] M. Naughton, N. Kushmerick, and J. Carthy, "Clustering sentences for discovering events in news articles," in *Advances in Information Retrieval, 28th European Conference on IR Research*, 2006, pp. 535–538.

[13] N. T. S. Sambasivam, "Advanced data clustering methods of mining web documents," *The Journal of Issues in Informing Science and Information Technology*, vol. 3, pp. 563–579, 2006.

[14] I. H. Witten, A. Moffat, and T. C. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images*. Van Nostrand Reinhold, New York, 1994.

On the Way towards Standardized Semantic Corpora for Development of Semantic Analysis Systems

Ivan Habernal and Miloslav Konopík
Department of Computer Science and Engineering
University of West Bohemia in Pilsen, Univerzitní 8, 306 14 Pilsen, Czech Republic
E-mail: habernal@kiv.zcu.cz, konopik@kiv.zcu.cz

Abstract—**One of the main means to achieve progress in science is cooperation. It is advantageous if the cooperation is carried among teams at different institutions. In semantics, the basic necessity for cooperation is a standardized annotated corpus. Such a corpus allows to share individual findings by the whole research community because then different systems can be tested under the same conditions. Unfortunately there is no standardized semantic corpus for the Czech language and many other languages suffer the same. Moreover the ATIS corpus set is more than ten years old and it does not meet today's trends in semantic annotation. In this article we summarize the problems of the ATIS corpora set as well as the problems encountered during our research. As a result, we provide a methodology to avoid such problems. For practical deployment of the methodology we offer a set of annotation tools. The purpose of this article is to discuss the problematic of semantic annotation and to gather other teams to create standardized shared semantic corpora.**

Keywords—**semantic analysis; semantic corpus; ATIS.**

I. Introduction

The goal of a Spoken Language Understanding system (SLU) is to extract a meaning from natural speech. The SLU covers many subfields such as utterance classification, speech summarization, natural language understanding (NLU) and information extraction. In human-computer dialogue systems, the task of the SLU system is to process the input acoustic utterance and transform it into a semantic representation. However, this task can be split into two parts: *automatic speech recognition* (ASR) and *semantic analysis*. The purpose of a semantic analysis system is to obtain a context-independent (it depends neither on history nor context) semantic representation from a given input sentence.

There are two basic types of semantic representation: *logical structures* (e.g., First-order predicate calculus, Transparent Intentional Logic, SIL, etc.) [1], [2] and *"data" structures* (e.g., trees, frames, flat concepts, etc.) [3]. The logical structures are more suitable for complex representation of semantics while the "data" structures are better suited for automatic learning systems. The reason is that statistical learning algorithms are not capable of handling the complexity of logical structures. Our experiences with semantic analysis systems based upon logical structures [2] shown that practical deployment of such systems is complicated due to the need of creating rules manually. Therefore, we focus on automatic learning systems, that seem to be more convenient for practical applications. Hence, we have chosen the tree based semantic representation described, i.e., in [3] that was designed mainly for practical use.

During the development and testing of the system described in [4], we have used our own Czech semantic corpus [5]. However, the results are not comparable with other semantic analysis systems since most of them (e.g., [6], [7]) performed their tests on different corpora. The availability of commonly used semantic corpora is quite good for English – for example the ATIS corpus [8], which is a mixed corpus for both speech recognition and semantic analysis. The tests on this corpus were performed by many semantic analysis systems. However, there is a lack of a standard semantic corpus for the Czech language, which differs from English in many aspects (morphologically rich, free word-order, etc.).

This paper presents our proposal to start the process of creation of such a corpus. It takes into account all practical issues that a developer of a semantic analysis system must deal with. It also describes the set of tools and proposes formats of the data. In this article we focus on the Czech language but most of the principles are valid for other languages too.

II. Related work

A. ATIS corpus

One of the commonly used corpora for testing of semantic analysis systems in English is the ATIS corpus. It was used for evaluation in, e.g., [6], [9], [10] and [11]. The original ATIS corpus is divided into several parts, e.g., ATIS2 train, ATIS3 train, two test sets, etc. [8]. Unfortunately, the corpus is not directly suitable for semantic analysis system development or testing.

The two testing sets, `ATIS3 test dec94` (445 sentences) and `ATIS3 test nov93` (448 sentences), contain the annotation in the semantic frame format. Each sentence is labelled with a goal name and slot names with an associated content. The training sets `ATIS2 train` and `ATIS3 train` contain only SQL queries that carry the semantic information.

This brings the first practical issue: To obtain the training data, the queries must be converted back to a semantic representation (a semantic frame or an equivalent semantic description). The authors of [6] transformed the data semi-automatically into a format suitable for the HVS model. Their training data use a bracketing notation to express the concept hierarchy. However, a deep exploration of this data shows

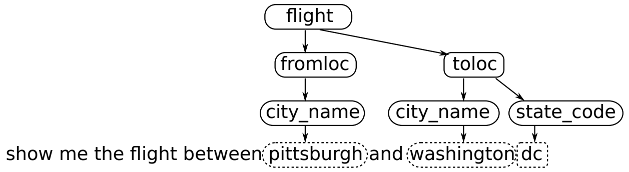


Fig. 1. An example of a semantic parse tree for a sentence from the ATIS corpus.

that a significant number of annotation break the conventions of the bracketing semantic annotation. The terminal semantic concepts (denoted as *lexical classes*) must be leaves of a semantic tree and are not allowed to contain any sub-tree (as shown in Fig. 1). However, in many cases the lexical classes act as superior concepts to other semantic concepts in the data. This inconsistency makes the data hard to use in other systems. This issue is, however, not caused by the ATIS corpus itself but re-creating the training data set from SQL queries probably always brings some sort of inconsistency.

Another issue is caused by inconsistencies in the testing data set. The following example shows a typical semantic frame for a flight query (this structure is equal to the semantic parse tree from Fig. 1).

```
show me the flight between pittsburgh and
washington dc
GOAL: FLIGHT
FROMLOC.CITY_NAME = pittsburgh
TOLOC.CITY_NAME = washington
TOLOC.STATE_CODE = dc
```

It has a very clear concept hierarchy. However, in the same testing set there also appears the following annotation:

```
what are the flight between dca and milwaukee
GOAL: FLIGHT
AIRPORT_CODE = dca
CITY_NAME = milwaukee
```

The semantic content of this sentence is rather similar to the previous one but the semantic frame is significantly different: In the second example, the concepts `AIRPORT_CODE` and `CITY_NAME` are directly inferior to the main concept (goal) without distinguishing which one is `FROMLOC` and `TOLOC`. Thus, the proper semantic frame should contain `FROMLOC.AIRPORT_CODE = dca` and `TOLOC.CITY_NAME = milwaukee`.

Another problem is how to deal with the annotation which has multiple goals. In the testing set there are about 20 sentences with two goals. The semantic interpretation part of a system (which is, in ATIS, a SQL query producer) should probably restrict the output of semantic analysis so that only one goal is allowed. Among others, there is also one typo in the testing data.

This brings two important questions: Were the testing sentences annotated according to any scheme? And how strictly was the testing set checked in a sense of inter-annotation agreement and correct semantic description?

B. Czech Semantic Corpora

Since the Czech language is morphologically rich and has a relatively free word order, it is not correct to directly adapt a semantic analysis system which is developed using an English corpus, and, obviously, a Czech semantic corpus is required. When searching for an existing suitable Czech corpus for the semantic analysis task, two significant projects must be mentioned.

The Prague Dependency Treebank (PDT 2.0) is a large corpus with morphological, syntactic and semantic (tectogrammatical) annotation. The methodology of adding the semantic layer to the PDT is described in [12]. The semantic representation formalism is based upon semantic networks and the tectogrammatical layer partially depends on syntax [13]. The tectogrammatical annotation provides a deep-syntactic (syntactical-semantic) analysis of the text. The formalism abstracts away from word order, function words (syn-semantic words), and morphological variation [14].

The DESAM corpus introduced in [15] was annotated with lemmas and gramatical categories. Subsequently, it was enriched with the semantic annotation [16]. The grammatical tagging was taken as a base and some tags were relabeled as semantic and pragmatic. The article [17] presents an attempt to combine Transparent Intensional Logic framework (which is used for capturing the semantics) with lexical units. Later, the semantic network (Czech WordNet) was enriched using morphological derivations [18].

However, the above mentioned corpora and related projects attempt to cover the semantics in a complex manner and are designed to act as a general description of semantics, in opposite to a task-oriented corpus such as ATIS.

Authors of [7] developed an extended HVS semantic parser (based on [6]) using a Human-Human Train Timetable Dialogue Corpus [19]. The corpus is annotated at multiple levels (dimensions) where the semantic dimension uses the same abstract annotation methodology as used in [6]. The corpus contains 1109 semantically annotated dialogues.

III. Standard Czech Semantic Corpus Requirements

A. The Task Definition

One of the main purposes of this paper is to inform and get the NLP and semantic analysis community involved into our task. It can be stated as: Creating a Czech semantic corpus, which will be publicly available, with clear and sufficiently universal semantic annotation structure, which is not limited to any domain. The corpus is not intended to describe the semantics as complex as presented in Section II-A but it should be strictly task-oriented, facing the practical issues that can arise during semantic analysis system development. Moreover, it will improve cooperation among the working groups focused on semantic analysis and will allow an objective comparison of the results.

B. Proposed Process Description

The proposed process workflow will consist of the following steps: First, a suitable text dialog corpus must be obtained. This can be based upn a part of the corpus presented in [5]. Second, an eligible semantic representation should be chosen. We discuss it in Section III-D. Third, the data will be annotated using semi-supervised learning and supporting tools presented in Section IV. Finally, to avoid the shortcomings that are for instance pointed out for the ATIS corpus, the annotated data will be manually validated.

C. Previous Work

Our attempt to create a semantically annotated corpus is presented in [5]. The semantic representation used in this corpus is based upon abstract semantic annotation from [6]. The corpus contains written user queries in natural language entered into an intelligent web search engine. A selected part of this data can be used as a basic set for the standard Czech semantic corpus.

D. Semantic Representation

To describe the semantics of an utterance, many task-oriented semantic analysis systems (e.g., [3], [7], [23], etc.) use some formats of the frame-based structure, as shown in the ATIS example. This simple formalism offers a very clear hierarchy of semantic concepts (a semantic tree), including the lexical realizations of the lexical classes. The name *lexical class* comes originally from [6], it can be also denoted as *named entity*, etc. It is a leaf of a semantic tree and covers one or more words with a specific meaning, such as names, dates, numbers, etc.

After considering the possible issues described in II-A, our previous corpus annotation effort and semantic analysis system development was supported by using an *annotation scheme*. The annotation scheme is a hierarchical structure (a tree) that defines a dominance relationship among concepts, theme (also called *goal* in ATIS or *topic*); this is the root semantic concept of the sentence. and lexical classes. It says which concepts can be associated with which super-concepts, which lexical classes belong to which concepts, and so on.

The annotation scheme should cover the entire domain we want to annotate. Subsequently, each sentence is annotated according to the scheme. The existence of such a scheme assures that two sentences with similar semantic content (meaning) will have the same semantic representation (see II-A). Apparently, this feature is crucial for further semantic interpretation.

However, the beforementioned annotation consistency using an annotation scheme is always limited to the covered domain. Althought this is not an issue for developers of a particular semantic analysis system, it does not allow to easily extend and evolve the scheme in the future together with maintaining the semantics of the annotation. Thus, it can be also considered to use more general formalism for describing semantics, i.e., RDF/OWL. Using this formalism, the corpus can be more easily aligned to other ontologies and then used in other semantic analysis systems with arbitrary semantic annotations. Furthermore, RDF/OWL has the same ability to prevent the annotators from creating malformed annotation as the annotation scheme which has proven to be essential for semantic corpus development [4].

IV. Supporting Tools

To improve the efficiency of the annotation [5] and to facilitate the corpus processing and sharing, supporting tools are required. We have developed a complete set of software covering the data acquisition, dialog act annotation and segmentation, semantic annotation and annotation management.

The first step of the data processing is conversion of a plain text into a format suitable for further annotation. This includes the text tokenization and morphological analysis (obtaining the morphological tags and the most probable lemma) using PDT 2.0. For this task, a web service has been developed and deployed.

The dialogue act segmentation is processed by the *dialogue act editor*. The output of dialogue act segmentation is then imported into the *abstract annotation editor*. The editor supports an advanced annotation methodology based upon automatic lexical class identification and bootstrapping. Both programs are GUI applications written in Java. The usability and efficiency of the tools has been presented in [20].

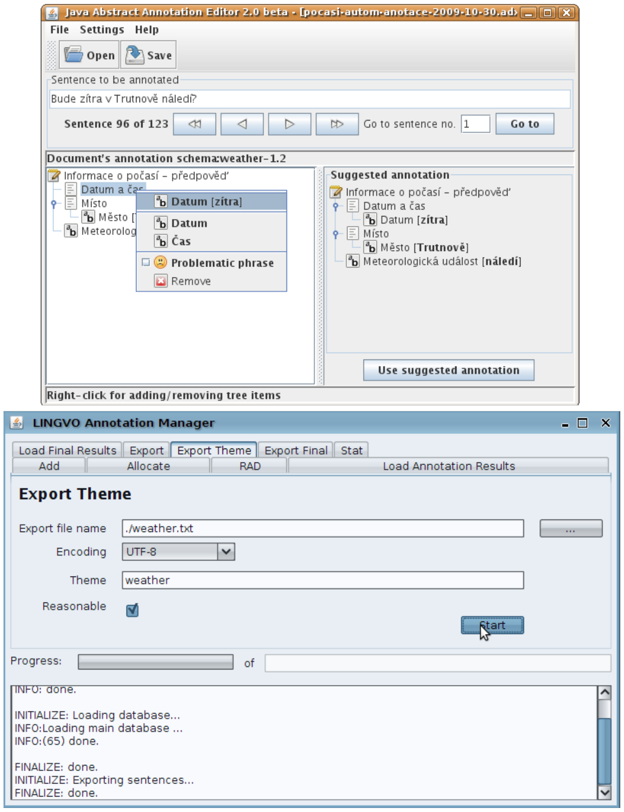
The *annotation manager* software helps to deal with an extensive semantic data. Some selected features are: A distribution of the sentences for the annotation among the annotators; annotation merging including conflict checking; various statistics (corpus statistics, annotation statistics, inter-annotation agreement, and annotator statistics). Again, this is a GUI Java application (see Figure 2).

All presented software tools are licenced under GPL licence and are publicly available from `http://liks.fav.zcu.cz`. At the same web page you can find information about the current state of the corpus, join the e-mail conference and get involved into the process of creating the standard Czech semantic corpus.

V. Conclusions

In this article, we have proposed an activity to create a standardized semantic corpus. We discussed issues that are connected with the annotation process of such a corpus. The basic parameters of the corpus to be created were described together with the process of how to create it. The expected impact of this article is to open a discussion of measuring the performance of systems for semantic analysis so that the results have an informative value.

Many recent articles about semantic analysis (e.g., [21], [22], [23], including ours) were published with the results measured on a private corpus. Our effort is to change this state by introducing a standardized semantic corpus. In order to be successful we, however, need a broad agreement on the details of the corpus to be created. That prevents us from creating such a corpus by ourselves and forces us to publish a work in progress.



Sentence to be annotated: Bude zítra v Trutnově náledí?

Sentence 96 of 123 Go to sentence no. 1

Right-click for adding/removing tree items

Export file name: ./weather.txt

Encoding: UTF-8

Theme: weather

Reasonable ☑

```
INFO: done.

INITIALIZE: Loading database...
INFO:Loading main database ...
INFO:(65) done.

FINALIZE: done.
INITIALIZE: Exporting sentences...
FINALIZE: done.
```

Fig. 2. A screenshots of the annotation editor and the annotation manager.

ACKNOWLEDGEMENTS

This work was supported by grant no. 2C06009 Cot-Sewing.

REFERENCES

[1] Allen J.: Natural Language Understanding, Benjamin/Cummings, Redwood City, CA, 1995.

[2] Ocelíková J., Matoušek V., and Krutišová J.: Design and implementation of a dialog manager. In: Text, Speech, Dialogue: Proceedings of the First Workshop on Text, Speech, Dialogue - TSD'98. Brno, Masaryk university. ISBN 80-210-1899-2, p. 257-262, 1998

[3] Young S.: The Statistical Approach to the Design of Spoken Dialogue Systems. Tech Report CUED/F-INFENG/TR.433, Cambridge University Engineering Department, 2002.

[4] Konopík M. and Habernal I.: Hybrid Semantic Analysis. In Proceedings of the 12th international Conference on Text, Speech and Dialogue. Lecture Notes In Artificial Intelligence, vol. 5729. Springer-Verlag, 2009, Berlin, Heidelberg, 307-314. ISBN 978-3-642-04207-2.

[5] Habernal I. and Konopík M.: Semantic Annotation for the LingvoSemantics Project. In Proceedings of the 12th international Conference on Text, Speech and Dialogue. Lecture Notes In Artificial Intelligence, vol. 5729. Springer-Verlag, 2009, Berlin, Heidelberg, 299-306. ISBN 978-3-642-04207-2.

[6] He Y. and Young S.: Semantic processing using the Hidden Vector State model. Computer Speech and Language, Volume 19, Issue 1, 2005, 85–106.

[7] Jurčíček F.: Statistical approach to the semantic analysis of spoken dialogues . Ph.D. thesis, p. 137, University of West Bohemia, Faculty of Applied Sciences, Pilsen, Czech Republic, Plzeň, 2007

[8] Deborah A. Dahl, et al., ATIS3 Test Data, Linguistic Data Consortium, Philadelphia, 1995

[9] Iosif E. and Potamianos A.: A soft-clustering algorithm for automatic induction of semantic classes. In *Interspeech-07*, pages 1609–1612, Antwerp, Belgium, August 2007.

[10] Jeong M. and Lee G.: Practical use of non-local features for statistical spoken language understanding. Computer Speech and Language, 22(2):148–170, April 2008.

[11] Raymond Ch. and Riccardi G.: Generative and discriminative algorithms for spoken language understanding. In *Interspeech-07*, pages 1605–1608, Antwerp, Belgium, August 2007.

[12] Novák V.: Semantic Network Manual Annotation and its Evaluation, Ph.D. thesis, Institute of Formal and Applied Linguistics, Charles University, Prague, Czech Republic, 129 pp., Sep 2008

[13] Cinková S.: Semantic Representation of Non-Sentential Utterances in Dialog, in Proceedings of SRSL 2009, the 2nd Workshop on Semantic Representation of Spoken Language, Athina, Greece, pp. 26-33, 2009

[14] Novák V., Hartrumpf S., and Hall K.: Large-scale Semantic Networks: Annotation and Evaluation, in Proceedings of the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions, Boulder, CO, USA, ISBN 978-1-932432-31-2, pp. 37-45, 2009

[15] Pala K., Rychlý P., and Smrž P.: DESAM - Annotated Corpus for Czech. In Proceedings of SOFSEM 97. Heidelberg : Springer Verlag, 1997. pp. 523-530. ISBN 3-540-63774-5.

[16] Pala K.: Semantic Annotation of (Czech) Corpus Texts. In Proceedings of the Second Workshop on Text, Speech and Dialogue. Berlin : Springer Verlag, 1999. pp. 56-61. Lecture Notes in Artificial Intelligence 1692. ISBN 3-540-66494-7.

[17] Pala K.: Word Senses and Semantic Representations. In Proceedings of the Third international Workshop on Text, Speech and Dialogue. Lecture Notes In Computer Science, vol. 1902. Springer-Verlag 2000, London, pp. 109-114. ISBN 3-540-41042-2.

[18] Pala K. and Sedláček R.: Enriching WordNet with Derivational Subnets. In Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing CICLING 2005. Springer Verlag, 2005. pp. 305-311, ISBN 3-540-24523-5.

[19] Jurčíček F., Zahradil J., and Jelínek L.: A Human-Human Train Timetable Dialogue Corpus. In Proceedings of EUROSPEECH, Lisboa, Portugal, 2005.

[20] Habernal I. and Konopík M.: JAAE: the Java Abstract Annotation Editor, In *INTERSPEECH-2007*, 1298-1301, 2007.

[21] Jurcicek F., Gasic M., Keizer S., Mairesse F., Thomson B. and Young S. Transformation-based learning for semantic parsing In: Proceedings of Interspeech 2009, 10th Annual Conference of the International Speech Communication Association, 6-10 Sept 2009, Brighton, UK.

[22] Wu W., Lu R., Duan J., Liu H., Gao F., and Chen Y. 2010. Spoken language understanding using weakly supervised learning. Comput. Speech Lang. 24, 2 (Apr. 2010), 358-382

[23] Zhou D. and He Y. 2009. Discriminative Training of the Hidden Vector State Model for Semantic Parsing. IEEE Trans. on Knowl. and Data Eng. 21, 1 (Jan. 2009), 66-77

SPARQL Compiler for Bobox

Miroslav Cermak, Jiri Dokulil and Filip Zavoral
Charles University in Prague, Czech Republic
{*cermak, dokulil, zavoral*}@*ksi.mff.cuni.cz*

***Abstract*—The Bobox framework is a platform for parallel data processing. It can even be used as a database query evaluation engine. However, it does not contain the means necessary to compile and optimize the queries. A specialized front-end is needed. This paper presents one such front-end, which handles queries written using the SPARQL language. The front-end also performs query optimizations taking the specific features of the SPARQL language into account.**

***Keywords*-SPARQL; Bobox; query optimization.**

I. Introduction

The SPARQL language [1] is one of the most popular RDF (Resource Description Framework [2]) query languages. There are several database engines that are capable of evaluating SPARQL. Unfortunately, their performance is still behind state-of-the-art relational and XML databases.

The Bobox parallel framework was designed to support development of data-intensive parallel computations [3]. One of the main motivation was to use it in web semantization research we are currently conducting [4]. The main idea behind Bobox is to connect large number of relatively simple computational components into a non-linear pipeline. This pipeline is then executed in parallel, but the interface used by the computational components is designed in such a way that they do not need to be concerned with the parallel execution – issues like scheduling, synchronization and race conditions. This system may be easily used for database query evaluation, but a separate query compiler and optimizer has to be created for each query language, since Bobox only supports a custom low-level interface for the definition of the structure of the pipeline.

Traditionally (for example in relational databases), query execution plans have the form of directed rooted trees where the edges indicate the flow of the data and all are directed to the root. The nodes of the tree are the basic operations used by the evaluation engine, like full table scan, indexed access, merge join, filter etc. This maps well to the Bobox archtecture, since the tree is a special case of the non-linear pipeline supported by the system. Each operation is mapped to one (or possibly a fixed combination of) components and the components are connected in the same manner as in the original evaluation plan.

We decided to take similar approach for SPARQL. An example of a query evaluation plan is shown in the Figure 1. While the operations used by the SPARQL algebra resemble operations used in the relational algebra, there are some more or less significant differences. This prevents the relational algebra from being used directly by the SPARQL evaluation engine. Furthermore, some optimizations used in relational optimizers are not applicable for SPARQL, since the several of the transformations that are used when optimizing relational queries do not preserve the semantics of the SPARQL queries.

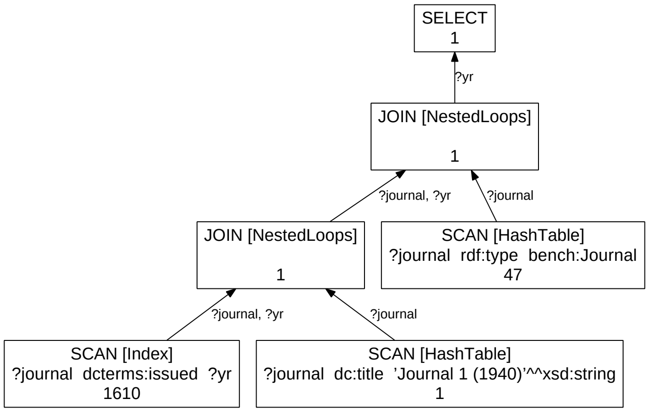


Figure 1. Query evaluation plan example

The rest of the text is organized as follows: first, the issues related to SPARQL execution, most notably the difference from relational algebra, are discussed in the Section II. Next, the Section III describes data representation used by our system and the statistics we collect about the queried data set. Sections IV and V describe the way in which the query is parsed, transformed and the way in which the final execution plan is generated from the transformed query. The Section VI provides some evaluation of our approach. The last section concludes the paper and discusses future work.

II. SPARQL

The sematics of the SPARQL language is defined using the SPARQL algebra. The algebra is similar to the relational algebra, but there are several important differences.

The relational algebra works with relations (tables), while the SPARQL algebra uses sets of variable mappings. Unlike SQL, there are no NULL values in SPARQL. Instead, the variable is left unbound. This is not just a minor technicality, it significantly affects the way in which some operations behave. There is no difference between an unbound variable and variable that is not present at all. For example the

SPARQL equivalent of left natural join produces a row (a variable mapping to be exact) even when the tested variable is unbound (which would also happen if the variable was not present in the query at all), unlike SQL where the operation is null-rejecting. This may prevent some optimizations like join-reordering to be performed on SPARQL since they could change the results under certain conditions. These optimizations may still be performed, but care must be taken to mind the specific constraints imposed by the SPARQL algebra.

There are two main circumstances when this behavior demonstrates. First, consider the following simple SPARQL query with a OPTIONAL expression:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?person, ?contact
WHERE {
      ?person foaf:mbox ?contact
      OPTIONAL(?person foaf:phone ?contact)
}
```

In this query, the contact variable is used both inside and outside the OPTIONAL pattern. The resulting behavior is that if the person has an email (foaf:mbox) the mail is returned but if the mail is not present, but a phone number is (foaf:phone), the phone number is returned. Performing this operation is SQL requires a more complicated combination of operations.

The second situation where different definition of left joins demonstrates is when the OPTIONAL branch contains FILTER operation. The operation cannot be performed on just the data from that branch – it is an integral part of the join operation and may filter even the data from the non-optional branch, as in the following example:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?person, ?contact
WHERE {
  ?person foaf:age ?x
  OPTIONAL(?person foaf:name ?b FILTER (?x>18))
}
```

On the whole, this means that we cannot directly apply optimization methods that were developed for SQL.

III. Data representation and statistics

Currently, we only work with local data sets. We assume that the data is stored in the most general model – one ”triple” table where all triples are stored. Besides that, we have several indexes, which are in fact the same table but with a specified order (using for example a B-tree). For example, one index is sorted by predicate, subject and the object. This may be used for example when we know the values of predicate and subject and we need to get all objects (and they are already sorted).

Besides the actual data and indexes, we need further information – statistics about data for the cost based optimizations. Since the number of different predicates is usually very limited, we can afford to store the number of distinct triples for each predicate. On the other hand, the number of subjects may be very high, so we only store the total number of triples and the number of distinct predicates. For the objects we use equal-height histograms [5]. This provides a balance between the size of the statistics and their precision – we do not store the number of triples for each object value, but if one value is much more common than the others, it would be detected from the histograms. These statistics allow us approximate selectivity of basic graph patterns.

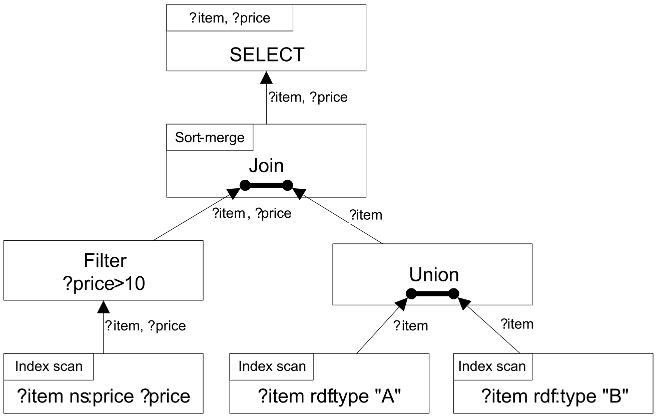


Figure 2. SQGM example

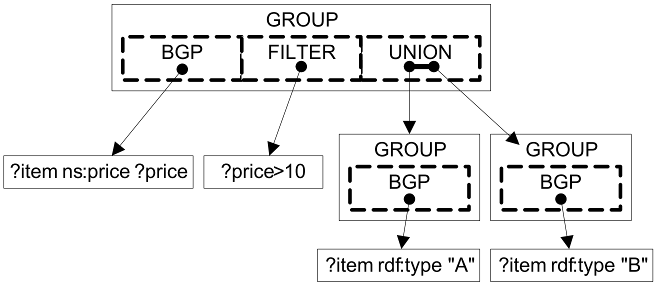


Figure 3. SQGPM example

We also try to make some approximations of the results of join operations. We consider the average selectivity of the join of two triples a_1, a_2, a_3 and b_1, b_2, b_3 with the join condition in the form $a_i = b_j$ – we only store one number for each possible form (9 combinations) of the join condition. Besides these general statistics, we use more detailed information for situations where both predicates are known. Then, there are four combinations on join conditions (a_1, p, a_2 and b_1, q, b_2 joined on $a_i = b_j$) for each pair of predicates. This means we have $4 * n^2$ values where n is the number of predicates. This should still be manageable amount of information.

During the optimization, the query evaluation plan is stored in the form of the SQGPM (SPARQL Query Graph Pattern Model) which we designed as an extension of the SQGM model [6]. The difference is that instead of individual operations, the nodes of the tree are formed by groups of operation – a group is a set of operations where the order in which the operations are evaluated does not affect the result of the operation. A SQGM model can be created by replacing each group of operations with a tree composed of those operations. This representation allows us to do transformations that are performed before the order of join operations is determined more easily, since the model is not yet made unnecessarily complicated by the "insignificant" joins (those whose order does not change the result of the query).

An example of the SQGM is shown in the Figure 2 and the corresponding SQGPM model in the Figure 3. They show models of the example query from the previous section.

IV. Query parsing and rewriting

The compilation of a query is performed in several consecutive steps. The first step is query parsing. The input stream is parsed into the SQGPM model using standard methods of lexical and syntactical analysis.

The next step is query rewriting. Since the queries to be processed are not expected to be written optimally (duplicities, constant expressions, inefficient conditions, etc.), the goal of this phase is normalization of the model. There are four operations performed on the SQGPM model:

- Merging of included *Group graph patterns*
- Duplicity elimination
- Propagation of `filter`, `distinct` and `reduced`
- Projection of variables

Resulting tree is functionally equivalent to the original one, but its evaluation can be more efficient and better execution plan can be generated.

A. Merging of included group graph patterns

The goal of the merging phase is to detect group graph patterns that can be merged with their parent group patterns while preserving the equivalence of the SQGPM.

Consider the following query:

```
SELECT *
WHERE { ?x ?a ?b . { ?x ?y ?z . { ?a ?b ?c } } }
```

There is only one possible operation ordering when the triples are grouped this way:

Join(?x ?a ?b, Join(?x ?y ?z, ?a ?b ?c))

Using such ordering the nested *Join* operation generates a cartesian product that is consumed by the outer *Join* operation. However, an equivalent representation of the query is as:

```
SELECT *
WHERE { ?x ?a ?b . ?x ?y ?z . ?a ?b ?c }
```

This representation results in wider range of operation ordering, e.g.:

Join(?x ?y ?z, Join(?x ?a ?b, ?a ?b ?c))

Such ordering does not produce the cartesian product; it uses smaller result sets and therefore is more efficient.

Nevertheless, not every *group graph pattern* (*GGP*) can be merged to its parent *GGP*. The problem arises if *GGP* contain both unbound variables and a *Filter* that defines restrictions on the variables. These variables may be bound in another *GGP*, in which case changing the scope of the *Filter* operation may change the result of the query.

Bound variables cannot change their value, they are safe with respect to the *FILTER* operation. The following example IV-A demonstrates a case where merging of *GGP*s is not possible:

```
SELECT *
WHERE {
      ?s rdf:type ?t .
      {P . FILTER(bound(?t)) }
}
```

P represents a graph pattern group for which the result set contains the variable `?s` and possibly unbound variable `?t`. Then the original representation rejects all tuples containing the unbound variable `?t` before joining the *triple* in the parent *GGP*. On the other hand, if we first join the nested *GGP* to the parent one and then perform the *Filter*, the variable `?t` will be bound by the parent *GGP* and the *Filter* never removes such result.

B. Duplicity elimination

The goal of the next phase is to eliminate duplicate graph patters. The following example demonstrates the problem:

```
SELECT DISTINCT *
WHERE {
   ?obj rdf:type ?t .
   ?obj rdf:type ?t
}
```

The query contains two equal triples `?obj rdf:type ?t`. The execution of the second triple and the subsequent join will not generate any new variable mapping not present originally.

If only bound variables are present, there is no combination of rows that would produce a new, unique row. The size of the result set is equal to the input set (possibly increased by duplicates). Then the `DISTINCT` modifier removes all duplicates which makes the result of the join equivalent to the original results of the `?obj rdf:type ?t` pattern.

This optimization may only be performed under the following conditions:

- Duplicate may not under any circumstances generate unbound variables
- The query is of the type `DISTINCT` or `REDUCED`

C. Propagation of Filter

Propagation of *Filter* means that we try to move it to the lowest level (closest to the leaves of the tree that represents the query plan) where all variables used in the *Filter* are still present. Early filtering reduces the size of the result sets which speeds up subsequent operations.

Operation *Filter* where the expression is in a conjunctive form is split into subexpressions using the operator AND. Such splitting reduces the expression domain (the set of the variables used in the *Filter*) and increases the probability of its lower placement in the resulting tree.

Nevertheless, the *Filter* operation cannot be propagated arbitrarily; presence of unbound variables prevents the propagation; see the following example.

```
SELECT DISTINCT *
WHERE { A . FILTER(bound(?y)) . { B }}
```

where:

- A, B are groups of operations with results:
A={{?x=1,?y=1}}
B={{?x=1}, {?x=2, ?y=2}}
- FILTER(bound(?y)) is a filter that uses the (possibly unbound) variable ?y

The result is the set {{?x=1, ?y=1}}. If we propagated the filter to the nested pattern, the result set would be empty. Therefore we defined *safe* and *unsafe* variables and conditions for *Filter* propagation:

- *Safe* variable is bound for every possible tuple
- *Unsafe* variable can be unbound for some tuple

If the *Filter*'s domain contains unsafe variables then it is ordered behind the last *group graph pattern* operation in the respective operation tree. If the *Filter* domain does not contain unsafe variables and it is not a part of a *group graph pattern* which forms the OPTIONAL branch of a *LeftJoin* then it:

- can be reordered behind the following operation (in a direction to the root)
- can be reordered before the preceding operation (in a direction to leaves) if it is not an OPTIONAL branch of the *LeftJoin* operation and all used variables are available.

The *Filter* operations that are part of the OPTIONAL branch of the *LeftJoin* operation cannot be reordered, since the SPARQL language defines it to be an integral part of the *LeftJoin* operation.

D. Propagation of Distinct and Reduced

If the query uses DISTINCT or REDUCED modifier, the result set should have no duplicates – they should be eliminated as the last step of the query evaluation. However, under most circumstances, we can add this operation even to deeper levels of the query plan, especially after *Join* (if it is a merge join) and *OrderBy* operations, since the data is ordered and the elimination of duplicates can be done very cheaply.

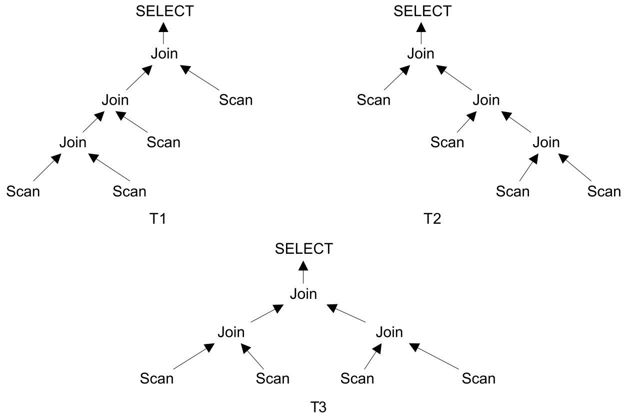


Figure 4. Tree types

V. Execution plan generation

After the transformations described in the previous sections, we still need to transform the groups present in the SQGPM model into a tree of join operations. This is performed by a non-exhaustive search of the space of all possible join types and combinations. We also have to select the best strategy to access the data stored in the physical store – select the best index to use, if any.

The query execution plan is built from bottom to the top using dynamic programming to search a part of the search space of all possible joins. We only consider left-deep trees of join operations, i.e. the right operand of a join operation may not be another join operation. See the Figure 4 for an example – T1 is a left-deep tree, T2 is right-deep and T3 is a bushy tree.

There is one exception to this rule. If there is no other way to add another join operation than adding one that would generate cartesian product, we try building the best plan for the rest of the operations (recursively using the same algorithm) and then join that plan with the one we already have. This may eliminate the need to generate cartesian products and results in an execution plan in the form of a bushy tree. This modification greatly improved plans for some of the queries we have tested and significantly reduced the depth of the trees – some of the results were almost a balanced binary tree.

The whole execution plan generation is performed by the following algorithm according to the statistics and price function that are able to provide an approximate cost for a part of any execution (sub)plan:

```
generate_plan(group_graph_pattern)
begin
  operators:=group_graph_pattern.childs;
  buckets:=empty;
  results:=empty;

  // Rating of feasible data access options
  foreach op in operators do
    foreach method in op.methods() do
      // Group operator recursive call
      if method is group then
        method := generate_plan(method);
      end if

      c := cost_of(method);
      s := sort_order_of(method);

      // The cheapest only
      if buckets[s].cost > c then
        buckets[s] := method;
      end if
    end for
  end for

  // Tree extension
  for i:=1 to |operators| do
    foreach tree in buckets do
      inops:=not used operators in tree;
      foreach op in inops do

        // Heuristics: skip the operators
        // that generate avoidable
        // cartesian products
        if carthesian && !required then
          continue;
        end if

        // Using join implementations
        foreach jtype in join_types do
          // Heuristics: left-deep tree
          newtree:= op_join(tree, op);
          c:=cost_of(newtree);
          s:=sort_order_of(newtree);

          if buckets[s].cost > c then
            buckets[s] := newtree;
          end if
        end for
      end for
    end for
    buckets:=res;
  end for

  // Result: the cheapest feasible plan
  return min from res;
end
```

The main goal of the design of this algorithm is to minimize the number of sort operations, make the best use of merge-join operations and avoid joins that generate cartesian products.

VI. Evaluation

An efficient implementation of the evaluation components for the Bobox system that could execute the generated query plans is not yet available. This allowed us to perform only two types of experiments so far: manually checking the plans generated by the compiler and comparing the cost estimates produced by the compiler with the actual size of the query result and intermediate results.

We have tested the queries provided by the SP^{2}Bench [7] benchmark suite for SPARQL. The Figure 5 shows an example the plan produced for the following query:

```
SELECT ?article
WHERE {
  ?article rdf:type bench:Article .
  ?article ?property ?value
  FILTER (?property=swrc:month)
}
```

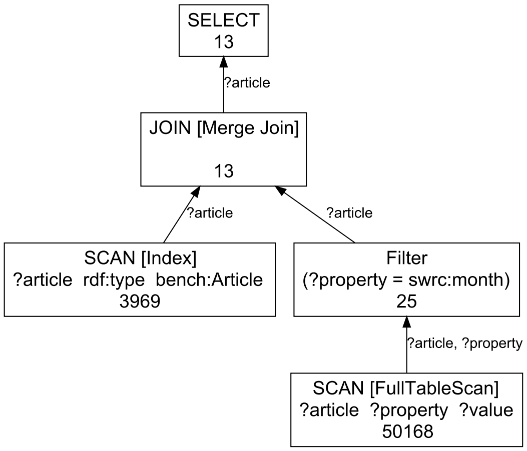


Figure 5. A simple query example

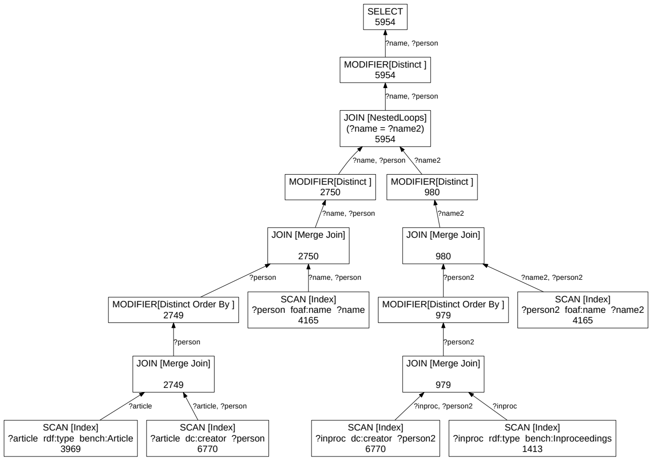


Figure 6. Example of a bushy tree

A more complex example that demonstrates the bushy trees that may be produced by the compiler is shown in the Figure 6. We were able to compile all `SELECT` queries defined by the SP^{2}Bench benchmark with satisfying results.

VII. Conclusion

We have created a working compiler that processes SPARQL queries and generates plans to be executed by the Bobox system. It performs a set of pre-defined optimizations to transform the execution plan into an equivalent but more efficient one. Then the query is further optimized by join reordering using dynamic programing and a cost model to asses the quality of the proposed execution plans.

An obvious next step is to implement the back-end of the SPARQL processor into Bobox and perform experiments on an actual physical RDF store. We have already created a subset of the back-end that can evaluate some of the SP^{2}Bench queries that have been compiled by hand to use only the specified subset of operations. The results of these experiments seem promising especially in comparison to current stat-of-the-art systems like Sesame [8].

Acknowledgment

The work was supported by the grants SVV-2010-261312 and GACR 202/10/0761.

References

[1] E. Prud'hommeaux and A. Seaborne, *SPARQL Query Language for RDF*, W3C, 2008, http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/. [Online]. Available: http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/

[2] J. J. Carroll and G. Klyne, *Resource Description Framework: Concepts and Abstract Syntax*, W3C, 2004, http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[3] D. Bednarek, J. Dokulil, J. Yaghob, and F. Zavoral, "Using methods of parallel semi-structured data processing for semantic web," in *3rd International Conference on Advances in Semantic Processing, SEMAPRO*. IEEE Computer Society Press, 2009, pp. 44–49.

[4] J. Dokulil, J. Yaghob, and F. Zavoral, "Trisolda: The environment for semantic data processing," *International Journal On Advances in Software*, vol. 1, no. 1, pp. 43–58, 2009.

[5] Y. Ioannidis, "The history of histograms (abridged)," in *VLDB '2003: Proceedings of the 29th international conference on Very large data bases*. VLDB Endowment, 2003, pp. 19–30.

[6] O. Hartig and R. Heese, "The SPARQL query graph model for query optimization," in *ESWC '07: Proceedings of the 4th European conference on The Semantic Web*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 564–578.

[7] M. Schmidt, T. Hornung, G. Lausen, and C. Pinkel, "SP2Bench: A SPARQL performance benchmark," *CoRR*, vol. abs/0806.4627, 2008.

[8] 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.

An Approach for Generating Semantic Annotations in Telecommunication Services

Simeona Harahap Cruz Pellkvist
University of Applied Sciences Technikum Wien
Höchstädtplatz 5, A-1200 Vienna, Austria
simeona.pellkvist@technikum-wien.at

Peter Reichl, Anna Fensel, Joachim Zeiss
FTW Telecommunications Research Center Vienna
Donau-City-Str. 1, A-1220 Vienna, Austria
{reichl | fensel | zeiss}@ftw.at

Abstract— **This paper describes the design and implementation of the "Semantic Generator" engine, which is used to transform and generate data from semantic formats, i.e., RDF, OWL or N3, to semantic and non-semantic formats, i.e., RDF/XML or text. The proposed lightweight approach maximizes the reuse of existing, widespread technologies while also allowing easy integration of new technologies. The generator engine's capacity is demonstrated and evaluated for two use cases with different requirements. On the one hand, it performs annotation generation of mobile service descriptions. On the other hand, the engine is used for mapping SIP messages from and to ontologies in real-time scenarios. In both cases, information available in a semantic format is mapped to the resulting semantic or non-semantic annotations and vice versa.**

Keywords - Semantic annotation, mobile services, microservice description, Session Initiation Protocol, IP Multimedia Subsystem.

I. INTRODUCTION

In the last years, semantic technologies have gained increasing interest in various fields of IT (Information technology), like Semantic Web and Business Process Management. Recently, there are also ongoing research efforts and projects on how these technologies can be leveraged in the field of telecommunications [1]. In particular, ontologies in the knowledge layers of telecommunication architectures play an increasing role for service platforms and mobile communications. As the integration of Telco, Internet and the Web takes place, in order to achieve interoperability, telecommunication systems and services tend to rely on knowledge represented with the use of shared schemata, i.e., on ontologies similar to those envisioned in the Semantic Web [2]. For example, when users move between different service domains, service delivery platforms might be necessary to dynamically change the service resources in order to provide the best user experience based on the context information or user preferences, e.g., by switching from one network operator to another, or between similar service components provided by the different service providers. Given efficient interoperability and semantic policies, it could be possible to substitute one service with another, if they can be proven to be sufficiently similar [3]. Therefore, as it is the case with services in general, semantic annotations will facilitate accurate service description, discovery and composition of telecommunications network services.

Among the many research projects in the engineering field using semantic technologies, we will focus on two particularly interesting ones, which are carried out at the Telecommunications Research Centre Vienna (FTW), Austria, i.e., m:Ciudad[1] and BACCARDI[2]. m:Ciudad addresses the question of how to create and share micro-services between the users of the mobile devices, where services should be created and shared to other users on the spot. In this project, service descriptions are generated out of information, which is available in semantic formats like RDF (Resource Description Framework) [4] and OWL (Web Ontology Language) [5] on the Web. For testing purposes, the service descriptions are generated automatically according to defined rules and schemata; and the users could share their own set up services to groups of friends or an organization. On the other hand, BACCARDI is an application-oriented research project with a broad focus on next generation fixed and wireless networks based on the Session Initiation Protocol (SIP) and using IMS (IP Multimedia Subsystem) as a testbed platform. One of the tasks in this context is to lift the header information of SIP `INVITE` and `BYE` messages to semantically described resources (i.e., RDF/N3 (Notation 3)). Furthermore, the transformation of semantically described resources to SIP route headers should be enabled to make round trips possible. Eventually, this allows for the execution of high-level policies on a technical level [6].

This paper is structured as follows: Section II presents the problem statement and a brief survey on relevant related work. Section III introduces our general approach and the Semantic Generator engine as our proposal to solve the problem. Section IV discusses the application of our prototype for the mentioned two case studies and presents results from our evaluation. Section V summarizes and concludes the paper with a brief outlook on the future work.

II. PROBLEM STATEMENT AND RELATED WORK

The need for a transformation process that translates semantic data to semantic or non-semantic formats (and vice versa) with some rules set up in between is common to both projects. On a higher level, such processes are generally needed

[1] m:Ciudad is a FP7 STREP that focuses on enabling end-user-generated mobile services. The project is running from 2007 to 2010, the consortium is comprised from 8 partners from several EU countries, coordinated by Robotiker-Tecnalia, Spain.
URI: http://www.mciudad-fp7.org.

[2] BACCARDI (Beyond Architectural Convergence: Charging, Security, Applications, Realization and Demonstration of IMS) is an Austrian COMET project which has been carried out at the Telecommunications Research Center Vienna from 2008 to 2010 in close collaboration with Telekom Austria, mobilkom austria, Alcatel-Lucent Austria, Kapsch CarrierCom and TU Vienna.
URI: http://www.ftw.at/ftw/research/projects/ProjekteFolder/COM-4.

for automatic generation of semantic annotations and service profiles, as well as for the purpose of testing and benchmarking large-scale search mechanisms. As large amounts of semantic annotations are time consuming and expensive to produce by hand, the suggested service annotation generation solution is of paramount importance, especially considering the tremendous growth of the semantically annotated resources. Thus, the main contribution of this work is an *approach and a prototype providing a configurable and flexible way to generate and transform telecommunication services annotations*.

Related work provides several frameworks and applications, developed and applied to transforming and generating services using semantic technologies. In [7], an approach is chosen to generate Web services automatically from a service graph model. An abstract model of services is created, and for this model a code generation is run to generate implementation files. However, this approach does not use semantic resources as an input. The approach chosen in [8] makes use of software agents to parse HTML (Hyper Text Markup Language) files and generates XML (eXtensible Markup Language) from them, enriching them with specific XML tags. Unfortunately it works on HTML only; therefore this approach is not applicable to our problem as well. The authors of [9] show how semantic data can be transformed to non-semantic data exploiting the capabilities of Model Driven Architecture in the domain of Business Process Engineering.

While these approaches work well in their specific domains, our goal is a more general solution, which can be used for a broad scope of applications and facilitates building user applications on proven traditional tools, while enabling an easy integration of arbitrary semantic schemata. In our solution, practically every semantic resource available on the Web could serve as an input. This allows crawling and extracting relevant information from large volumes of interlinked ontologies and semantic annotation resources, particularly, the whole Linked Open Data Cloud [10].

III. The Semantic Generator Engine

A. Approach

While currently there are no ready to use applications available that can solve the entire transformation problem introduced previously, there are at least some frameworks available, which can be leveraged to fulfill parts of the requirements. Therefore, we combine these frameworks in order to use the advantages of each tool and integrate them to an engine that produces the required results.

More specifically, the transformations addressed in our work involve:

- Querying information from semantic resources,
- Querying information from SIP messages,
- Combining the data that has been extracted, and
- Formatting and splitting the data according to the requirements.

The engine is required to be able to read the semantic resources – in our case, RDF/XML, N3, and OWL – and also to write, at least in RDF/XML format (for the m:Ciudad case) and in text/N3 format (for the BACCARDI case). Moreover, the tool must able to reformat and split combined data according to given rules. In addition, querying should be performed similar to the well-known and standard SQL syntax in order for achieving an easy and familiar usage. The extracted data should be able to be combined with or without any repetition.

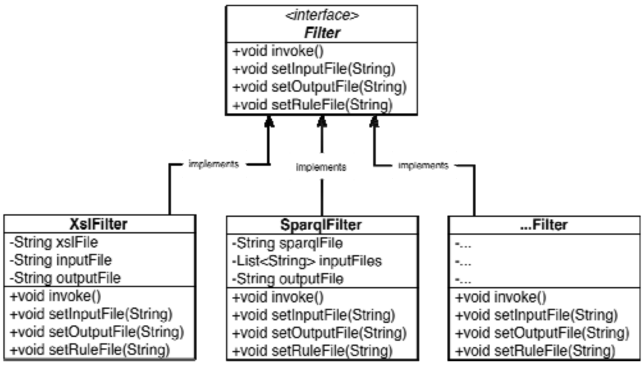
For an efficient employment of the available tools they have to be combined using design patterns [11][12], which are well known from software design. In general, a design pattern is a template, which could be applied for several situations in a certain condition in order to achieve a general solution for a number of common problems, which happen repeatedly. In order to achieve this we use “Pipes and Filters Pattern” and “Adapter Pattern”.

In software engineering, Pipes and Filters usually mean that the output or result of one application is used as input for another one. The Pipes and Filters pattern in our case represents an architectural pattern for the overall application. Different forms of the inputs are passed to the first filter and are processed there. This result is transferred to the second filter for which it represents the input. This process repeats to the next filter and so on. Therefore, this method is suitable to transform the semantic data, in our case into an RDF/XML..

The Adapter pattern is also known as wrapper pattern (or wrapper) and describes a technique used to make classes, which have different interfaces that are compatible to each other. An adapter may also be used to convert data to a suitable format. In our case, the adapter is used to wrap external resources; in our case, XSLT (eXtensible Stylesheet Language Transformation) and SPARQL (SPARQL Protocol and RDF Query Language).

B. Main Engine

The main generator engine is the core integration layer built upon the Pipes and Filters and the Adapter pattern. It represents an abstraction mechanism for the other frameworks, provided they have serializable input and output, and rules for controlling this transformation. This is achieved by implementing the common Filter interface. Fig. 1 shows the important core classes of the generator, specifically the filter interface that is implemented for example by the SPARQL filter and the XSLT filter.



<interface>
Filter
+void invoke()
+void setInputFile(String)
+void setOutputFile(String)
+void setRuleFile(String)
implements
implements
implements
XslFilter
-String xslFile
-String inputFile
-String outputFile
+void invoke()
+void setInputFile(String)
+void setOutputFile(String)
+void setRuleFile(String)
SparqlFilter
-String sparqlFile
-List<String> inputFiles
-String outputFile
+void invoke()
+void setInputFile(String)
+void setOutputFile(String)
+void setRuleFile(String)
...Filter
-...
-...
-...
+void invoke()
+void setInputFile(String)
+void setOutputFile(String)
+void setRuleFile(String)

Figure 1. Class diagram (excerpt)

1) *Semantic Query Filter*

Two semantic query filters have been implemented: the SPARQL filter based on ARQ [13] using the Jena [14] toolkit, and a SeRQL (Sesame RDF Query Language) based filter using the Sesame [15] toolkit. The SPARQL based filter is our main filter for querying semantic resources in this engine, whereas the SeRQL based filter is provided for comparison to the SPARQL based filter, playing an important role for the engine has and involving three main steps:

- First, the filter takes the semantic files, such as RDF file, N3 file, OWL file, as the input.
- Using specific sets of rules, the input files are then stored where the query is stored, either in SPARQL or in SeRQL.
- Finally, the output is an XML file from the results of the SPARQL and SeRQL query language. Both of them use the XML result format from SPARQL such that they can be easily replaced by each other.

2) *SIP Query Filter*

In this filter, the engine is assigned to separate the SIP messages into different fields for reasoning purpose. The input of the SIP query filter consists of SIP messages, which are in text format. The algorithm translates the SIP messages into N3 serialization to be passed to a reasoner, which is supposed to infer information over semantically annotated data, based on logical rules. In our engine, this serialization can theoretically be set to another format file, depending on which rule is set. The input is read from the `InputStream` and saves the result as a string. JAIN API [16] is used to parse the string into a SIP message, before the default Jena model is initialized. The filter is set to find out if the SIP message is a SIP request or a SIP response. When the message is already defined, its content then is separated to different fields as Subject - Predicate - Object "sentences" in the Jena model under the condition that the related field is not null. The message is handled differently according to the SIP request or SIP response. The result is written to an `OutputStream`.

3) *Combining Filter*

The purpose of this filter is to combine the queried output results from the semantic filter. The input for this filter consists of the output results that are written in an XML file format from a semantic query filter. This filter has different sets of rules for different purposes of output result; the rules can be defined as requested. The filter combines the rules and orders the elements, generates a maximum of results and optionally considers uniqueness. The output of this filter is in XML file format. The filter reads the config file, where it can be specified if every input shall only be used once. Furthermore, this filter will take care of randomizing the results. The output will be written as a result in XML.

4) *Formatting Filter*

The output result of the previous filter is not formatted yet with respect to the user's design, hence this procedure is done in this filter. Two formatting filters have been implemented: one based on XSLT, another one based on XQuery (XML Query Language).

IV. Case Studies

Having outlined our general approach in Section III, we will now discuss two case studies where the suggested transformation approach has been applied, i.e., the research projects m:Ciudad project and BACCARDI.

A. *m:Ciudad*

As already mentioned earlier, the m:Ciudad case study is focusing on semantic descriptions of mobile micro-services. For this purpose, a tool is required, which is able to transform available semantic data written in RDF or OWL file form to XML file form.

During the transformation, the data of semantic descriptions in the form of RDF or OWL should be extracted, combined/integrated, and reformatted to different RDF formats according to the requirements. As input and output formats may change, the transformation should be able to be adapted in a flexible way.

In order to create mobile service descriptions, RDF and OWL files constitute the input for our engine. These inputs are queried and combined using the SPARQL filter over the Linked Open Data Clouds [17], and the result is persisted to an XML file. The output file of the first filter then becomes the input file for the XSLT filter, which has again XML files as output. This process could be developed to filters that are piped, to support additional input or transformations, i.e., the parsing/writing of SIP messages or replacing XSL transformations by another XML transformation language.

The implementation of this engine, which queries RDF- and OWL-based knowledge bases, generates any required number of various service annotation datasets required for our testing and benchmarking purposes. These datasets are compliant with the m:Ciudad's schemata of the Service Profile, i.e., the basic annotation of a service, and the Service Capability, i.e., annotation of basic service behavior and requirements. Each element has been formulated in different fields according to what has been set on the rule. The number of created datasets is large enough to be used, particularly, for performance evaluation of the micro-service employment algorithms.

Figure 2. and the subsequent listings illustrate the m:Ciudad solution in more detail. **Listing 1** represents one of the input files, i.e., the bloggers.rdf, which is an RDF store conforming to the FOAF (Friend of a Friend) ontology. **Listing 2** shows the configuration file that drives the transformation process. **Listing 3** shows the configuration rule for the first filter in the pipe, i.e., the SPARQL filter. An SQL-like query asks the name from the FOAF ontology. In this query, abbreviations for namespaces are defined using the keyword `'PREFIX'`. Variables begin with a `'?'`. The query listed here basically translates to: Find an entity `x` that has a `foaf:name`, bind this name to the variable `CapabilityName` and return it. **Listing 4** shows the intermediate result for the capability name. **Listing 5** shows the configuration file of the combining filter as a result of the SPARQL query above. In order to reformat the information and structure it according to a service profile, an XSL stylesheet is used, which is shown in **Listing 6**. As the intermediate result contains all the result rows in one file, it is necessary to split them in different files; this is achieved using a Xalan specific instruction (xalan:write) for every `sparql:result` element. For every sparql:binding, one element is created in the example. E.g., if the binding is for `CapabilityName`, a `udlcp:Capability` element is created. An attribute is created for it, which has an attribute `datatype`, whose value is set to string (using the corresponding XML schema datatype). The value of the `sparql:literal` or `sparql:uri` element in the input is put as content of the created element. Finally, **Listing 7** shows the resulting service capability. The capability name derives from the foaf:name in the foaf ontology.

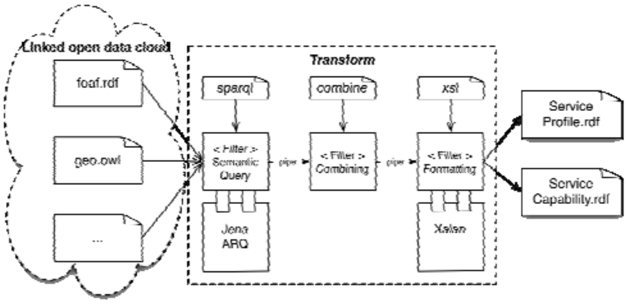


Figure 2. mCiudad solution

```
<foaf:Agent rdf:nodeID="ni93487857">
<foaf:name> Pasta N Pizzor </foaf:name>
<foaf:weblog>
 <foaf:Document
rdf:about="http://www.example.com/pastapizzor">..
```

Listing 1. Input file: Friend-of-a-Friend (FOAF) RDF

```
<generatorConfiguration>
<workflow name="CapabilityWF">
<filter class="id.shc.genie.SparqlFilter">
<input>http://danbri.org/foaf.rdf</input>
<output>../work/CapabilityName.xml</output>
<rule>../etc/CapabilityName.sparql</rule>
</filter>...
```

Listing 2. Transform: Workflow Configuration file

```
PREFIX foaf : <http://xmlns.com/foaf/0.1/>
SELECT ? CapabilityName
WHERE { ?x foaf :name ?CapabilityName . }
```

Listing 3. SPARQL filter configuration

```
<sparql>
<head> <variable name="CapabilityName"/> </head>
<results><result>
<binding name="CapabilityName">
<literal xml:lang="en"> Pasta N Pizzor </literal>
</binding></result>...
```

Listing 4. SPARQL Profile result

```
COMBINE 100 :
CapabilityID unique
CapabilityName
CapabilityDescription
...
```

Listing 5. Combining filter configuration

```
<stylesheet version="1.0"
  extension-element-prefixes="xalan"
...
<output method="xml" encoding="ISO-8859-1" in-
dent="yes" />
...
<template match="sparql:result">
<xalan:write se-
lect="concat('capability-',position(),'.rdf')">
<udlcp:Capability> <apply-templates />
</udlcp:Capability>
</xalan:write>
</template>

<template match="sparql:binding">
<if test="@name = 'CapabilityName '">
<udlcp:CapabilityName>
<attribute name="datatype"namespace="&rdf;#">
&xsd;#string
</attribute>
<value-of select="sparql:literal|sparql:uri"/>
</udlcp:CapabilityName>
</if>
...
```

Listing 6. Formatting filter configuration: XSL Stylesheet

```
<udlcp:Capability
xmlns:udlcp="http://www.mciudad-
fp7.org/schemas/udlcp#">
  <udlcp:CapabilityID>59</udlcp:CapabilityID>
  <udlcp:CapabilityName>Pasta N Piz-
zor</udlcp:CapabilityName>
  ...
```

Listing 7. Output file: Service Capability RDF/XML

Summarizing, the engine supports two different languages for semantic queries, based on two different frameworks and two different XML transformation languages, in order to investigate how the flexibility of the engine is able to cope with different applications.

B. BACCARDI

In the BACCARDI use case, semantic web based policy definitions are enabled, which compose and control application services hosted in an IMS core network. In the BACCARDI Service Oriented Data-driven Architecture (BACCARDI SODA) working group, a N3-based semantic reasoner, the so-called "policy engine" [17][18], is used to make semantic policy-based decisions on how to combine or modify behavior of application services, which communicate via SIP or ISC (IMS Service Control), respectively (ISC is an extension of SIP for call and service control purposes in IMS). Instead of talking to the application services directly, the SODA architecture proposes to intercept SIP `INVITE` and `BYE` messages to cancel, redirect or manipulate message information based on policy decisions of the reasoner, and thus to compose and control service behavior in real time.

In the BACCARDI case, SIP messages have to be transformed to N3 statements, which are subsequently sent to the reasoner over HTTP. The answers received from the reasoner in terms of N3 have to be parsed, and accordingly SIP headers are manipulated.

For this purpose, SIP messages have to be intercepted by a SIP proxy servlet. These messages have to be translated to N3, for which a special filter is needed. As this is the only transformation step, it shall be able to directly embed the filter into the corresponding servlet, without having to use the engine for setting up the transformation. Figure 3. depicts the basic solution for the BACCARDI case, while Figure 4. shows how servlet container, SIP proxy servlet, SIP2N3 filter and the reasoner service work together.

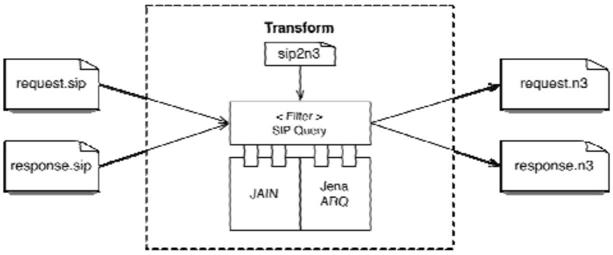


Figure 3. BACCARDI solution

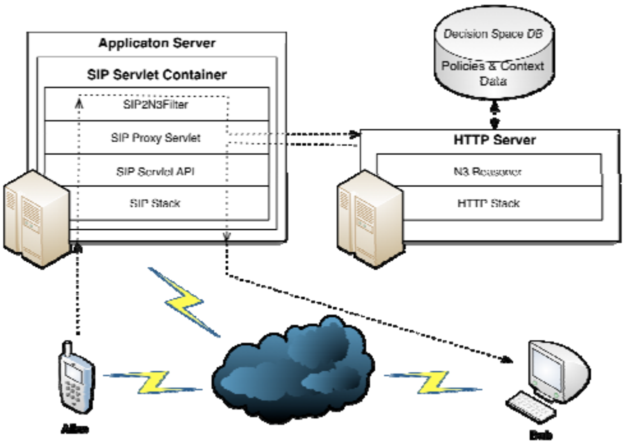


Figure 4. Setup for Baccardi

For the BACCARDI use case, the result has been applied to different scenario episodes. The result is shown in log files, demonstrating the reaction of the engine to the scenarios that have been set up. As a specific example, we now consider the “Forward” scenario, which is applied to check when the SIP user agent makes a call and this call is then forwarded to another participant. Suppose the caller is defined as Alice and the callee is defined as Bob, while Charlie is define as the one whom the call shall be forwarded to. **Listing 8** shows how the original log file of the SIP INVITE message looks like. This SIP request is translated to N3 when passing the engine, and forwarded to the reasoner. As we can see in **Listing 9**, the SIP INVITE message has been translated to N3 for reasoning purposes. The reasoner answers with "FORWARD" using Charlie’s username/number. **Listing 10** shows the response of the reasoner mock, and **Listing 11** shows the modified forward SIP INVITE message, where Bob’s phone doesn’t ring, instead Charlie gets Alice’s calls. Charlie answers the call and the answer 200 OK message from the forward number, see **Listing 12**.

During our practical experiments, we observed that the engine gives proper response and translates the original messages to N3 for reasoning purposes. In the BACCARDI SODA case, the implementation of this engine has successfully transformed the header information from SIP INVITE and BYE messages to semantic annotated data instances according to the BACCARDI SODA ontology. The header information from SIP INVITE and BYE messages are transformed to N3; also SIP route headers can be manipulated, based on the results of the N3 based reasoner according to the BACCARDI SODA ontology, in order to send, cancel or forward SIP messages to other application servers.

```
INVITE sip:Bob@128.131.202.184:22244 SIP/2.0
Max-Forwards: 70
Content-Length: 255
To: "Bob"<sip:Bob@128.131.202.184:22244>
Contact: <sip:Alice@128.131.202.184:62901>
Cseq: 1 INVITE
Content-Type: application/sdp
From: "Alice"<sip:Alice@test.com>;tag=684e0942...
```

Listing 8. Scenario Forward: Original SIP invite message

```
<request> sip:contact
"<<sip:Alice@128.131.202.184:62901>>" .
<request> sip:cSeq 1 .
<request> sip:protocol "SIP/2.0" .
<request> sip:content_type "application/sdp" .
<request> sip:request_url
"<sip:Bob@128.131.202.184:22244>" .
<request> sip:max_forwards 70 .
<request> sip:to
"sip:Bob@128.131.202.184:22244" .
<request> sip:content_length 255 .
<request> sip:from "sip:Alice@test.com" .
<request> a        sip:INVITE .
...
{<response> soda:action ?a} => [] .
{<response> soda:add_header ?b} => [] .
{<response> soda:delete_header ?c} => [] .
{<response> soda:append_value ?d} => [] .
```

Listing 9. Scenario Forward: SIP invite message translated to N3

```
<response> soda : action soda : forward . <re-
sponse> soda:forward_address
"sip:Charlie@128.131.202.184:51267".
```

Listing 10. Scenario Forward: Reasoner answer

```
INVITE sip:Charlie@128.131.202.184:51267 SIP/2.0
Max-Forwards: 70
Content-Length: 255
To: "Bob"<sip:Bob@128.131.202.184:22244>
Contact: <sip:Alice@128.131.202.184:62901>
Cseq: 1 INVITE
Content-Type: application/sdp
From: "Alice"<sip:Alice@test.com>;tag=684e0942...
```

Listing 11. Scenario Forward: Modified SIP invite message

```
SIP/2.0 200 OK
Record-Route:
<sip:128.131.202.184:5060;lr;fid=server_1>
Content-Length: 253
```

```
To:
"Bob"<sip:Bob@128.131.202.184:22244>;tag=126fb82d
Contact: <sip:Charlie@128.131.202.184:51267>
Cseq: 1 INVITE
Content-Type: application/sdp
From: "Alice"<sip:Alice@test.com>;tag=684e0942...
```

Listing 12. Scenario Forward: 200 OK message

An emulation of the reasoner has been developed to test if the SIP proxy servlet and filter reacts properly to the SIP messages. It is implemented as an HTTP servlet, and allows the configuration for different scenarios, i.e., the unchanged forwarding or redirection of a SIP message, the cancellation of the related SIP dialog or the manipulation of header files in the SIP message based on the response of the reasoner. Both, the SIP proxy servlet and the BACCARDI SODA "policy engine" represent a semantic IMS SCIM (Service Capability Interaction Manager), which controls services across application servers in the IMS network.

V. Conclusions and Future Work

In this paper we have demonstrated how to successfully apply semantic technologies for scalable and flexible data transformation and generation within a single prototypical engine, i.e., the Semantic Generator. The chosen approach and the corresponding solution has been validated and tested in two different case studies from two ongoing telecommunications industrial projects.

For the m:Ciudad case, we have successfully queried semantic resources from the Web, aggregated and combined the results, and transformed them, thus generating mobile service descriptions according to different configurable set of rules. The engine was able to successfully generate 10,000+ datasets of service profiles and capabilities. In the BACCARDI case, we have transformed the different SIP messages to N3 for reasoning purposes. The developed filter has been embedded in a SIP proxy servlet, and this approach has been evaluated for different test scenarios. The results have been documented by screen capturing and log files, which show that the chosen approach and the developed solution fulfill the requirements concerning functionality.

It could be argued if the development of a common architecture has really been necessary, or if two different architectures for solving the two problems should be preferred. While it is not necessary to use the same architecture, this approach has certain advantages, for example it extends maintainability and enables combination of filters. For instance, the SIP2N3 filter cannot only be used in a real-time scenario like in the BACCARDI project, but also to enable lifting of information from log files for enabling reasoning or for providing input to service generation. Further, it allows a step-by-step and bottom-up style of development, thus reducing the entry barrier to semantic technologies for users who are currently using traditional technologies. The generator engine enables them to continue to use their proven tools while facilitating and promoting the use of semantic technologies.

Summarizing, the proposed approach has been successfully evaluated for different applications. The developed engine is flexible, and its behavior can be changed easily by adapting configuration files. Furthermore, the extensible architecture of the engine also allows the user to create their own filters according to their needs with reasonable effort, which underlines once more the efficiency of our solution.

Acknowledgment

Part of this research has been funded by the Austrian Government and the City of Vienna within the competence center program COMET.

References

[1] A. V. Zhdanova, N. Li, and K. Moessner: Semantic Web in Ubiquitous Mobile Communications. The Semantic Web for Knowledge and Data Management (Ed.: Ma, Z.), IGI Global, August 2008.

[2] S. Tarkoma, C. Prehofer, A.V. Zhdanova, K. Moessner, and E. Kovacs: SPICE: Evolving IMS to next generation service platforms. In: Proceedings of the 3rd Workshop on Next Generation Service Platforms for Future Mobile Systems (SPMS 2007) at the 2007 International Symposium on Applications and the Internet, IEEE Computer Society Press, 2007.

[3] T. van Do and I. Jorstad: A service-oriented architecture framework for mobile services. In: Proceedings of Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop AICT/SAPIR/ ELETE, 17-20 July 2005, pp. 65 – 70.

[4] http://www.w3c.org/RDF, 2009-08-24, 2010-02-09.

[5] G. Antoniou and F. van Harmelen. Web Ontology Language: OWL. Springer-Verlag, 2003.

[6] A. V. Zhdanova, J. Zeiss, A. Dantcheva, R. Gabner, and S. Bessler: A Semantic Policy Management Environment for End-Users and its Empirical Study. Volume 221/2009 of Studies in Computational Intelligence. Springer Berlin / Heidelberg, 2009.

[7] E. Cho, S. Chung, and D. Zimmerman. Automatic Web Services Generation. In HICSS, pages 1–8. IEEE Computer Society, 2009.

[8] D. Camacho and M. D. R-Moreno: Web Data Extraction using Semantic Generators. In: International e-Conference of Computer Science (IeCCS 2006), LNCS, pp. 34–38, 2007.

[9] C. Blamauer and D. M. R. Lintner: Integrating Semantic Business Process Management and Viewbased Modelling. Master's thesis, Vienna University of Technology, 2009.

[10] C. Bizer, T. Heath, and T. Berners-Lee: Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems, Special Issue on Linked Data, 2009.

[11] E. Gamma, R. Helm, J. Vlissides, and I. R. Johnson: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.

[12] F. Buschmann, R. Meunier, H. Rohnert, and P. Sommerlad: Pattern-oriented Software Architecture - A System of Patterns. J. Wiley and Sons Ltd., 1996.

[13] http://jena.sourceforge.net/ARQ/documentation.html, 2010-02-04.

[14] http://jena.sourceforge.net/, 2010-02-04.

[15] http://www.openrdf.org/doc/sesame2/2.3.1/users/index.html, 2010-02-09.

[16] https://jain-sip.dev.java.net/, 2010-01-08.

[17] J. Zeiss and O. Jorns: Using Semantic Reasoning and Privacy Policies in Ubiquitous Envi- ronments. UBICOM2007 Workshop: First International Workshop on Security for Spontaneous Interaction, IWSSI 2007, Innsbruck, Austria, 16th September, 2007.

[18] S. Bessler and J. Zeiss: Using Semantic Policies to reason over User Availability. Second International Workshop on Personalized Networks, Pernets'07, Philadelphia 2007.

Cosema: Content-based Semantic Annotator

Angela Fogarolli
University of Trento
Via Sommarive 14, 38100 Trento, Italy
afogarol@disi.unitn.it

***Abstract*—In this paper, we present a library for creating automatic annotations for entities and concepts inside any textual content. The tool is based on DBpedia. In particular, the annotations are generated using the DBpedia link structure as a source of knowledge for Word Sense Disambiguation. DBpedia is used as a reference to obtain information on lexicographic relationships. By using such information in combination with statistical information extraction techniques, it is possible to deduce concepts related to the terms extracted from a corpus. Moreover, by combining statistical information extraction with named entity recognition and the use of the OKKAM ENS infrastructure, it is also possible to obtain unique annotations for entities in the content. The advantage of this approach, in addition of improving information retrieval and categorization capabilities, consists in the fact that the generate concept and entity annotations can be referred to with unique identifiers around the Web. For this reason different description for the same entity or concept can be semantically aggregated from the Web.**

I. Introduction

A common practice to avoid information overloading is to enable efficient access to a resource by associating to documents a set of metadata which describe their content. These metadata usually provide additional information about the content of the resources they are describing, such as author, main topic, language, etc. Descriptions should have a high level of semantics in order to be used for answering human needs of classification and retrieval. Various standardized metadata descriptors, which fulfill these requirements to different extents, are available today.

Metadata can be manually generated this is costly, time consuming and can be error-prone. Also the agreement between annotators can notably differ and usually requires domain expertise and controlled vocabularies. Since the amount of documents people are dealing with is constantly increasing, manual annotation faces increasing challenges in terms of sustainability. However, knowing what a document is about is of fundamental importance for effective knowledge management. Automatic or semi-automatic techniques can be employed instead as an alternative to human annotation. The limitation of automatic annotation is usually low recall when annotations are missing, low precision when the annotations are inaccurate, or the extraction of relationships [1] among them. Additionally, annotations alone do not establish the semantics of the vocabulary used in the annotations.

A solution to this problem can be inspired by the Semantic Web. The Semantic Web as envisioned in [2] allows semantic interoperability between machines and users. It provides a stack of languages for supporting the representation of knowledge, in the form of ontologies and metadata. Semantic Web technologies aim to annotate documents based on domain Ontologies. In this way the semantics of the produced annotations are well defined. An ontology [3] is a conceptualization of a domain with a controlled vocabulary and grammar for describing objects and the relations between them in a formal way. Ontologies are populated with individuals, often referred to as (named) entities. Typical entities are specific (individual) people, organizations, events, artifacts (“Mona Lisa”), places, products, etc.

The vision of the Semantic Web involves re-use mainly of the schematic parts of ontologies, i.e. concepts and their definition. Uniform Resource Identifiers (URIs) are used for referring to any resource, relations between resources can be stated in RDF [4] statements, and the vocabulary used for describing these relations is specified using RDF Schema [5] or OWL citeowl ontologies. The benefit of using this kind of formalization is that information can rather easily be aggregated (by detecting identical URIs in datasets), and that they enable certain kinds of reasoning (e.g., about class hierarchies) that can produce query results beyond what is currently possible using relational databases or information retrieval systems.

The environment described in this paper aims to provide a way for automatically generating semantic annotations for a given text compliant with best practices of the Semantic Web, being easy interlinking and distributedness. We thus enable extraction and sharing of the knowledge implicit in content, on the Web. For guaranteeing domain independence, the tool is based on the DBpedia [7] knowledge. DBpedia can be considered a light weight ontology which spans different domains. In this way, any type of content can be annotated by Cosema library.

The rest of the article is organized as follows. In the next section we will give a brief overview of the related work.

In Section III we describe a novel approach for automatic generation of concepts and entities semantic annotations. In Section IV we will evaluate the quality of the automatic generated annotations. The conclusions summarize our contribution to the Semantic Annotation field.

II. Related Work

Semantic Web technologies aim to automatically or semi-automatically annotate documents based on domain ontologies. In this way the semantics of the produced annotations are well defined. Semantic annotations define in a formal way concepts and relationships between them. There are different approaches from manual to automatic generation of annotations. In [1] a review of the state of the art in the field is presented.

The use of annotations has been investigated in various fields. Examples include: television and radio news [8], bioinformatics [9], heritage [10] and content classification of web pages [11] Human annotation is costly, time consuming and prone to errors. Also the agreement between annotator can notably differ and requires domain expertise.

Cucerzan in [12] described an interesting approach for associating Name Entities in a corpus with Wikipedia definitions. The goal of this approach is similar to ours, the main difference is that we do not limit the corpus analysis to Name Entities and we considered also multilanguage material. They explored various strategies to decrease the numbers of attributes to consider. They reduce the context information by extracting entities with a certain number of mentions in the article or using some TF-IDF threshold. For learning about topic dependencies for annotation, in this paper we consider only strong links [13] among articles.

Synarcher [14] is another work based on Wikipedia knowledge, which searches for synonyms and related terms in the Wikipedia category structure and analyzing hyperlinks between pages. The algorithm could be used to extend queries in a search engine, or as an assistant for forming a dictionary of synonyms. Another work which explores categories in Wikipedia is the one of Chernov et al. [15]. The authors suggest that semantic information can be extracted from Wikipedia by analyzing the category structure and they propose a way to calculate a connectivity ratio which correlates with the strength of the semantic connection among them. Wikipedia categories are also used for document classification by Schonhofen [16] and by Thom et al. [17] for improving entity ranking effectiveness. Watanabe et al. present another work on Name Entity categorization [18] based on category information extracted from the linked HTML text in the articles. Syed et al. in [19] describe an approach for identifying topics and concepts associated with a set of documents. The approach is based on the Wikipedia category graph for predicting generalized concepts and uses article links to help predict concept when an article is not associated with a specific category.

Adafre and de Rijke [20] firstly analyzed the link structure in Wikipedia, in 2005. They tackle the problem of missing links between articles. For doing this they cluster similar pages based on similar link structure and then they examined these cluster to find missing links between them. Voss [21] described the Wikipedia link structure as a power law function in which there is an exponential growth of links. Whenever a non-existing article is linked is more likely someone will create it. Kamps and Koolen [22] examined Wikipedia link structure and stated that link structure is an indicator of relevance especially if considering links between pages retrieved in response to a search request. In other words links can help defining a context and can improve performance in information retrieval. Hyperlinks structure in Wikipedia is also used for calculating related pages to an article. Ollivier and Senellart [23] process these relationships using Green Measures which is a function introduced in electrostatic theory for computing the potential created by a charge distribution. Green measures are applied as a finite Markov chain to a graph modeled by hyperlinks among Wikipedia articles.

Mihalcea in [24] and [25] discuss the use of Wikipedia for Word Sense Disambiguation (WSD). In [24], the author reports about the use of Wikipedia content for avoiding the bottleneck in WSD of not having enough examples of a term usage. In her approach, she selects all paragraphs in Wikipedia which contain a contextualized reference to an ambiguous term in the link label and then maps the different Wikipedia annotations to word senses instead of relying on the Wikipedia disambiguation pages. This is due to the face that sometimes not all meanings are elicited in the disambiguation page. Finally, the labels which describe the possible senses for a word are manually mapped to WordNet senses. In this way the number of example for each word can increase improving the performance of a classifier. In her second work [25], Mihalcea describes an use case of her WSD algorithm to an application which associate terms in an input text to Wikipedia definitions. The keyword extraction from the text is done using a controlled vocabulary. WSD is done in three different ways. Using a Knowledge-Based calculating the overlap of the Wikipedia definition with the paragraph where the text occurs (similar to Lesk algorithm). A second approach that has also been tested in [25] is a data-driven method which use a machine learning classifier, giving as a training all the occurrences where the word is found in the link plus all the possible Wikipedia definition articles, which represents the possible meanings. Additionally they experimented also a combination of the first two approaches.

The OKKAM research project [26] is an attempt to solve the identity problem on the Semantic Web. OKKAM aims to enable and bootstrap the Web of Entities, a global decentralized information space in which every entity is identified by a global identifier, and in which global identifiers are consistently used for specifying relations between entities, across system boundaries. As the World Wide Web (WWW) was the result of integrating local Webs of documents into a global (universal) space of resources addressable through global identifiers (the well-known URLs), so the Web of Entities will be the result of integrating local webs of entities (i.e. any local space of information about a collection of entities, like a directory, a catalogue, an information system, a knowledge base, a database, a data intensive web site, and so on) into a global information space where every entity is identified through a global (universal) identifier. However, with respect to the WWW, the domain of entities is extended beyond the realm of digital resources, and links between entities are extended beyond hyperlinks to include virtually any type of relation. As a result, the vast amount of information, which today is not integrated, could be aligned and become part of a global information space that has entities as pivot objects, instead of documents.

III. Generation of Semantic Annotations

In this section, we describe how we extract semantic annotations from textual content. Those annotations express the most important concepts and entities in text content. There are two interfaces for accessing the functionality of the Cosema library, a web interface and a Web service interface. As input the system receives a text passage and it returns semantic annotations for contained entities and concepts. The annotations are represented by using Semantic Web URI. In this way by resolving the URI is possible to gather a detailed description of the meaning of the annotation.

A. *Disambiguation Process*

This section describes the WSD process we used for discriminating the correct meaning of a term based on the context where the term was found. The approach is based on the DBpedia link structure which can be assimilated to the Wikipedia link structure. The link structure in Wikipedia draws a huge network between pages which facilitates the navigation and the understanding of concepts.

The type of link we are interested in for WSD are what we called "strong links". We define a strong link as a bidirectional connection between two pages. A page P_o has a strong link with page P_d if in P_o exists a link to P_d and in P_d there is a link back to P_o.

$$P_o \longleftrightarrow P_d \tag{1}$$

A link in Wikipedia is considered to be strong if the page it points to has a link back to the starting page.

The WSD approach included in the Cosema library uses DBpedia as a source of Knowledge.

The first step for calculating semantic annotations is related to information extraction (IE). Cosema uses two IE methods: a statistical one based on TF-IDF measure and a name entity recognizer (NER) (two commercial and one opensource NER has been evaluated).

A term vector containing the most important terms on a document is extracted based on the TF-IDF measure and combined with the results of the NER. The disambiguation for an ambiguous term or entity is calculating by matching the term with a DBpedia or OKKAM identifier. The process takes into account the document domain which is defined by the terms in the same document.

In Wikipedia and so in DBpedia, different word senses are represented through a so-called disambiguation page. Each article in Wikipedia is identified by its title. The title consists of a sequence of words separated by underscores. When the same concept exists in different domains that name is concatenated with a string composed by a parenthetical expression which denotes the domain where the word has a specific sense. If a query ambiguously identifies more senses, a disambiguation page is called.

The algorithm for creating a semantic annotation uses two different resources for annotating entities or concepts. For entity annotations Cosema relies on the knowledge of OKKAM ENS which already includes all the DBPedia entities. While it uses DBpedia directly for disambiguating concepts and in the case of entities that are not present in the OKKAM ENS. It follows a separate description for the two methodologies. The results of the IE phase are two lists, one with the extracted entities derived from the NER and the second is a term vector coming from the statistical IE. Each of the extracted entities is looked up in OKKAM. In case of entity type "'Person"' there is the need of minimum two words (since the ambiguity of using just a last name or a name as discrimination will be too high) to be passed to OKKAM otherwise the entity will be resolved with the procedure used for the concepts which deals with ambiguity by taking into account the context where the word was located. If the entity is present in OKKAM then OKKAM identifiers and the entity alternative identifiers will be returned (i.e., the DBpedia identifier can be an alternative identifier).

For generating annotations for concepts or for entities in case of failure of the OKKAM lookup, the procedure will analyze every term present in the term vector created out of the text given as input. The term vector is defined as:

$$T_{i=1..N} = \{w_{ij}\}_{j=\{1..25\}}$$

where i identify a specific document, and j a term in the term vector. For each candidate definition p_{ijk}, where k is the k-th possible definition, we consider only its strong links (the concept and its links are searched in DBpedia through the SPARQL endpoint).

$$S_{zijk} = S_z\left(p_{ijk}\right)_{z=\{1..M\},k=\{1..Q\}} \tag{2}$$

where Q is the number of senses for pij and M is the number of strong links for the k-th sense.

Therefore S_{zijk} is the z-th strong link for the k-th sense of the j-th term of the i-th document. Hence, a strong link represents a bidirectional relation between two DBpedia pages. All strong links S_{zijk} for every term w_{ij} are taken into account for computing the disambiguation process and to be used in the query suggestion and summarization task. The best definition among the candidates is the one having the majority of words w_{ij} in the presentation material T_i in common with the target article name anchored from a strong link.

We can write this concept as function $f(i,j,k)$ where i identifies a specific document, j a term in the term vector and k a candidate definition for the term j. The function $f(i,j,k)$ will help us selecting the page p_{ij} which has the maximum number of elements in the intersection between the term vector for a presentation T_i and the target article name of the selected hard links for the candidate DBpedia definition pages, p_{ijk}. The function $f(i,j,k)$ is defined as:

$$f(i,j,k) = |T_i \cap \left\{S_{zijk,z=\{1..M\}}\right\}|$$

where z is the i-th strong link for the candidate page p_{ijk}.

The symbol | indicates the cardinality of the expression. The correct definition page p_{ij} will be identified among the p_{ijk} pages by selecting the k such that $|f(i,j,k)|$ has the largest value.

$$p_{ij} = p_{ijk}$$

which indexes are found by

$$max_k|f(i,j,k)|$$

For example if we analyze an e-Learning document (document 1) about Java Programming whose (simplified) vector is defined by:

$$T_1 = \{set, map, array, list, java, computer, collection\}$$

We consider the case of finding the right DBpedia definition for the term collection which is part of document 1. In the disambiguation page are listed the definitions for ”Collection(computing)” and ”Collection(museum)”. For each of these pages we analyze the strong links counting the number of elements in common with the words in the term vector of the e-Lecture document in exam:

$$S_{171}Collection\,(computing) =$$

$$\{oriented, class, map, tree, set, array, list\}\,;$$

$$S_{172}Collection\,(museum) = \{curation, curator\}\,;$$

The group CE, contains the elements in common between the term vector and the strong link for each candidate page:

$$CE = T_1 \cap S_{171}Collection\,(computing)$$

$$CE = T_1 \cap S_{172}Collection\,(museum)$$

Since words in a term vector are stemmed, the strong links must be stemmed as well before comparing them with the keywords in the term vector. We choose the DBpedia definition page among the candidate pages to be the one which has the maximum number of elements in CE. In the example, we have $|f171| = 3$ (case of $Collection\,(computing)$) while $|f172| = 0$ (case of $Collection\,(museum)$). Therefore the disambiguated meaning of term P_17 (i.e. collection) is correctly found to be $Collection\,(computing)$. The expected result of the process is a complete disambiguated term vector Td_i composed of disambiguated words wd_{ij}.

$$Td_{i=1..N} = \{wd_{ij}\}_{j=\{1..25\}}$$

For improving the accuracy of the results we do not insert in the candidate pages only the ones with an exact match to a word in the analyzed text but all the pages which begin with that word. In this way, we are sure to include in the candidates definitions all the declinations and possible domain. More specifically, there are cases where ambiguous words are not linked to the articles mentioned by a disambiguation page, but instead they are mentioned in the related concepts section or a disambiguation page does not exist.

The disambiguation process access DBpedia online through the Web service interface, while other approaches presented also in the related work section use Wikipedia directly. The major drawback of using Wikipedia instead of DBpedia is that Wikipedia is not structured and there is not API for automatically accessing its content. For this reason for accessing Wikipedia content there is the need of using Natural Language processing techniques directly on the online version with very poor processing performance or installing and using the Wikipedia dumps. The dumps supply a complete database with the Wikipedia content; the drawbacks of this solution are in maintaining and keeping the local Wikipedia copy up to date for then calculating semantics on it. Using DBpedia instead is a very fast, lightweight and always up to date alternative for collecting information about Wikipedia content.

IV. Evaluation

Assessing the quality of an application is very difficult and depends highly on human expertise. We evaluated the quality of the described approach in WSD. The idea behind our approach is based on a link analysis of DBpedia definition pages. In, our previous work [13], we supplied evidence that since links among Wikipedia pages connect articles that are semantically related and likely on the same context, the link structure also provides a way for identifying relationships among topics. Furthermore, we want to investigate how strong these relationships are, based on the type of link that exists between the documents. In particular, we suppose that if there is a symmetrical link relationship among two

Evaluation type	Precision
Wikipedia Based Corpus Tagging	73.4%
DBpedia Based Corpus Tagging	76.1%

Table I
WSD COMPARISON

pages, the strength of the link denotes the most important connections for describing a subject. In this section we want to evaluate how good is the approach in creating semantic annotations and for doing this we have to focus on evaluating the WSD task at the base of our approach.

The objective of the evaluation is to assess the quality of the system in recognizing different word sense. In this section we want to explore if the approach can produce good annotations for describing the content of generic text.

For this purpose we have collected sixteen text passages in English, from various sources: newspapers, encyclopedia, text books and random Web pages. We asked two annotators to manually annotate the passages using three titles of Wikipedia articles for restricting the vocabulary possibilities. Next, we compared the annotations automatically generated with the manual ones using two testers. The testers after careful reading of each text passage had to judge the correctness of the automatic annotations taking into account the difference in semantics between them and the manual ones by expressing a quality value from zero to one. A zero quality value means that the automatic annotations does not describe the text passage and they are completely unrelated with the manual annotations and one means that the automatic annotation perfectly describe the text content and can be the same as the manual annotations. We let the testers free to autonomously decide the other values in the interval by judging the semantic error of the automatic annotations.

In order to calculate the result of the experiment we consider the manual annotation to be exact and we compare the automatic ones against them. Based on the two tester judgment the precision on our test collection of the automatic generated annotations is 75%.

This result is consistent, as shown in table I, with a previous evaluation we made using Wikipedia dumps for calculating WSD. This underlines that for concept disambiguation the information included in the DBpedia representation is sufficient for gathering the same accuracy results as with Wikipedia. This result support our assumption that DBpedia knowledge can be used as Wikipedia for creating semantic annotations with the advantage of a faster processing time and easier accessibility.

During the word sense evaluation we were also considering the correctness of the meaning of the annotation by pointing it to a Wikipedia article, for this reason the precision value is lower since some errors can occur in the sense disambiguation while the annotation word is still correct. For example for a text about the “9/11” an annotation Attack could be consider correct but the meaning of the connected article given by our algorithm was “Attack (30 Seconds to Mars song)” which is wrong. In the WSD evaluation the objective was to have both correct annotation and sense, on the automatic tagging evaluation the focus was only on the correct annotation. Moreover in the test we only compared the results for concept annotations and not entities annotations since our previous Wikipedia based approach was not able to distinguish between entities and concepts. Even though we do not present this type of comparison, the persons who took the test admit that the entity annotations where able to give either an higher level categorization of the text in case of events or a more specific definition in case of person entities.

V. CONCLUSIONS AND FUTURE WORK

We have presented a library tool for automatic generation of concepts and entities annotations about content. The library can be accessed through a Web interface or Web Services. In the paper the WSD approach behind the tool is described and evaluated. DBpedia has been used as a knowledge resource for WSD. The cross-links between DBpedia entries allows us to discover important relations between concepts. We applied the presented work in a digital library environment for automatically annotating and enabling searches and navigation through an unstructured multimedia and in another tool for creating multimedia presentation. The good results of the evaluation suggest that our approach might be applied in different scenarios such as text categorization and document classification, where it is crucial to automatically extract semantic information from content. This underlines the genericity and usefulness of the work presented in this paper. In the future we plan to add the functionality of generating an RDFa description of the annotations to be included where the content will be published. In this way semantic search engine will easily discover the annotated content.

REFERENCES

[1] V. S. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna, “Semantic annotation for knowledge management: Requirements and a survey of the state of the art,” *J. Web Sem.*, vol. 4, no. 1, pp. 14–28, 2006.

[2] T. Berners-Lee, J. A. Hendler, and O. Lassila, “The Semantic Web.” *Scientific American*, vol. May, 2001.

[3] D. Fensel, *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003.

[4] P. Hayes, *RDF Semantics*, February 2004, http://www.w3.org/TR/rdf-mt/. [Online]. Available: http://www.w3.org/TR/rdf-mt/

[5] D. Brickley and R. G. (Eds.), *RDF Vocabulary Description Language 1.0: RDF Schema*, W3C, February 2004.

[6] P. Patel-Schneider, P. Hayes, and I. Horrocks, "Web Ontology Language (OWL) Abstract Syntax and Semantics," W3C, Tech. Rep., February 2003, http://www.w3.org/TR/owl-semantics/. [Online]. Available: http://www.w3.org/TR/owl-semantics/

[7] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "Dbpedia : a crystallization point for the web of data," 2009. [Online]. Available: http://jens-lehmann.org/files/2009_dbpedia_jws.pdf

[8] M. Dowman, V. Tablan, H. Cunningham, and B. Popov, "Web-assisted annotation, semantic indexing and search of television and radio news," in *WWW '05: Proceedings of the 14th international conference on World Wide Web*. New York, NY, USA: ACM, 2005, pp. 225–234.

[9] E. Camon, M. Magrane, D. Barrell, D. Binns, W. Fleischmann, P. Kersey, N. Mulder, T. Oinn, J. Maslen, A. Cox, and R. Apweiler, "The gene ontology annotation (goa) project: implementation of go in swiss-prot, trembl, and interpro." *Genome Res*, vol. 13, no. 4, pp. 662–672, April 2003. [Online]. Available: http://dx.doi.org/10.1101/gr.461403

[10] V. M. Hennie Brugman and L. Hollink, "A common multimedia annotation framework for cross linking cultural heritage digital collections," in *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, E. L. R. A. (ELRA), Ed., Marrakech, Morocco, may 2008.

[11] S. Handschuh and S. Staab, "Authoring and annotation of web pages in cream," in *WWW '02: Proceedings of the 11th international conference on World Wide Web*. New York, NY, USA: ACM, 2002, pp. 462–473.

[12] S. Cucerzan, "Large-scale named entity disambiguation based on wikipedia data," in *EMNLP 2007: Empirical Methods in Natural Language Processing,Prague, Czech Republic*, June 28-30, 2007, pp. 708–716. [Online]. Available: http://acl.ldc.upenn.edu/D/D07/D07-1074.pdf

[13] A. Fogarolli, "Word sense disambiguation based on wikipedia link structure," in *IEEE ICSC 2009*, 2009.

[14] A. Krizhanovsky, "Synonym search in wikipedia: Synarcher," *arxiv.org*, search for synomyms in Wikipedia using hyperlinks and categories. [Online]. Available: http://arxiv.org/abs/cs/0606097v1

[15] S. Chernov, T. Iofciu, W. Nejdl, and X. Zhou, "Extracting semantic relationships between wikipedia categories," in *1st Workshop on Semantic Wikis:*, June December 2006.

[16] P. Schonhofen, "Identifying document topics using the wikipedia category network," in *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 456–462.

[17] A.-M. Vercoustre, J. A. Thom, and J. Pehcevski, "Entity ranking in wikipedia," in *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*. New York, NY, USA: ACM, 2008, pp. 1101–1106.

[18] Y. Watanabe, M. Asahara, and Y. Matsumoto, "A graph-based approach to named entity categorization in Wikipedia using conditional random fields," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 649–657. [Online]. Available: http://www.aclweb.org/anthology/D/D07/D07-1068

[19] Z. Syed, T. Finin, and A. Joshi, "Wikipedia as an ontology for describing documents," in *Proceedings of the Second International Conference on Weblogs and Social Media*. AAAI Press, March 2008.

[20] S. F. Adafre and M. de Rijke, "Discovering missing links in wikipedia," in *LinkKDD '05: Proceedings of the 3rd international workshop on Link discovery*. New York, NY, USA: ACM, 2005, pp. 90–97.

[21] J. Voss, "Measuring wikipedia," in *Proceedings International Conference of the International Society for Scientometrics and Informetrics: 10 th*, 2005. [Online]. Available: http://eprints.rclis.org/archive/00003610/

[22] J. Kamps and M. Koolen, "The importance of link evidence in wikipedia." in *ECIR*, ser. Lecture Notes in Computer Science, C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven, and R. W. White, Eds., vol. 4956. Springer, 2008, pp. 270–282. [Online]. Available: http://dblp.uni-trier.de/db/conf/ecir/ecir2008.html#KampsK08

[23] Y. Ollivier and P. Senellart, "Finding related pages using Green measures: An illustration with Wikipedia," in *Proc. AAAI*, Vancouver, Canada, Jul. 2007, pp. 1427–1433.

[24] R. Mihalcea, "Using wikipedia for automatic word sense disambiguation," in *Proceedings of NAACL HLT 2007*, 2007, pp. 196–203. [Online]. Available: http://www.cs.unt.edu/~rada/papers/mihalcea.naacl07.pdf

[25] R. Mihalcea and A. Csomai, "Wikify!: linking documents to encyclopedic knowledge," in *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. New York, NY, USA: ACM, 2007, pp. 233–242. [Online]. Available: http://84.11.13.37/Volumes/CIKM_07/docs/p233.pdf

[26] P. Bouquet, H. Stoermer, C. Niederee, and A. Mana, "Entity name system: The back-bone of an open and scalable web of data," *International Conference on Semantic Computing*, vol. 0, pp. 554–561, 2008.

Recognizing Textual Entailment with Deep-Shallow Semantic Analysis and Logical Inference

Andreas Wotzlaw and Ravi Coote
Fraunhofer Institute for Communicaton, Information Processing and Ergonomics FKIE
Neuenahrer Str. 20, 53343 Wachtberg, Germany
Email: $\{$*andreas.wotzlaw, ravi.coote*$\}$*@fkie.fraunhofer.de*

***Abstract*—In this paper, the architecture and evaluation of a new system for recognizing textual entailment (RTE) is presented. It is conceived as an adaptable and modular environment allowing for a high-coverage syntactic and semantic text analysis combined with logical inference. For the syntactic and semantic analysis it combines an HPSG-based deep semantic analysis with a shallow one supported by statistical models in order to increase the quality and accuracy of results. For recognizing textual entailment we use logical inference of first-order employing model-theoretic techniques and automated reasoning tools. The inference is supported with problem-relevant background knowledge extracted automatically and on demand from external sources like, e.g., WordNet, YAGO, and OpenCyc, or other, experimental sources with, e.g., manually defined presupposition resolutions, or with general and common sense knowledge. The system comes with a graphical user interface for control and presentation purposes. The evaluation shows that the success rate of the presented RTE system is comparable with that of the best logic-based approaches.**

***Keywords*-recognizing textual entailment; semantic analysis; logical inference; knowledge integration; semantic reasoning.**

I. Introduction

In this paper, we present a new system for *recognizing textual entailment* (RTE, see [1], [2]). Our aim is to provide a robust, modular, and highly adaptable environment for a linguistically motivated large-scale semantic text analysis. In RTE we want to identify automatically the type of a logical relation between two input texts. In particular, we are interested in proving the existence of an entailment between them. The concept of *textual entailment* indicates the situation in which the semantics of a natural language written text can be inferred from the semantics of another one. RTE requires a processing at the lexical, as well as at the semantic and discourse level with an access to vast amounts of problem-relevant background knowledge [3]. RTE is without doubt one of the ultimate challenges for any natural language processing (NLP) system. If it succeeds with reasonable accuracy, it is a clear indication for some thorough understanding how language works. As a generic problem, it has many useful applications in NLP [4]. Interestingly, many application settings like, e.g., information retrieval, paraphrase acquisition, question answering, or machine translation can fully or partly be modeled as RTE [2]. Entailment problems between natural language texts have been studied extensively in the last few years, either as independent applications or as a part of more complex systems, e.g., during the RTE Challenges [2].

In our setting, we try to recognize the type of the logical relation between two English input texts, i.e., between the text T (usually several sentences) and the hypothesis H (one short sentence). More formally, given a pair $\{T, H\}$, our system can be used to find answers to the following, mutually exclusive conjectures with respect to background knowledge relevant both for T and H [5]:

1) T entails H,
2) $T \wedge H$ is inconsistent, i.e., $T \wedge H$ contains some contradiction, or
3) H is informative with respect to T, i.e., T does not entail H and $T \wedge H$ is consistent.

We aim to solve an RTE problem by applying a *model-theoretic* approach where a formal *semantic representation* of the RTE problem, i.e., of the texts T and H, is computed. However, in contrast to *automated deduction* systems [6], which compare the atomic propositions obtained from the text and the hypothesis in order to determine the existence of entailment, we apply *logical inference of first-order*. To compute semantic representations for input problems, we build on a combination of deep and shallow techniques for semantic analysis. The main problem with approaches processing the text in a shallow fashion is that they can be tricked easily, e.g., by negation, or by systematically replacing quantifiers. Also an analysis solely relying on some deep approach may be jeopardized by a lack of fault tolerance or robustness when trying to formalize some erroneous text (e.g., with grammatical or orthographical errors) or a shorthand note (e.g., short text message). The main advantage when integrating deep and shallow NLP components is increased robustness of deep parsing by exploiting information for words that are not contained in the deep lexicon [7]. The type of unknown words can then be guessed, e.g., by usage of statistical models.

The semantic representation language used for the results of the deep-shallow analysis is a first-order fragment of *Minimal Recursion Semantics* (MRS, see [8]). However, for their further usage in the logical inference, the MRS expressions are translated into another, semantic equivalent representation of *First-Order Logic with Equality* (FOLE) [5]. This logical form with a well-defined model-theoretic semantics was already successfully applied for RTE in [9].

As already mentioned, an adequate representation of a natural language semantics requires access to vast amounts of common sense and domain-specific world knowledge. RTE systems need problem-relevant background knowledge to support their proofs [3], [10]. The logical inference in our system is supported by external background knowledge integrated automatically and only as needed into the input problem in form of additional first-order axioms. In contrast to already existing applications (see, e.g., [2], [9]), our system enables flexible integration of background knowledge from more than one external source (see Section IV-A for details). In its current implementation, our system supports RTE, but can also be used for other NLP tasks like, e.g., large-scale syntactic and semantic analysis of English texts, or multilingual information extraction.

In the remainder of the paper, we give first a short overview of related work (Section II). Then we present in detail the architecture of our system (Section III) and explain how its success rate can be improved by employing external knowledge and presupposition resolvers (Section IV). The paper concludes with a discussion of the results (Section V).

II. Related work

Our work was inspired by the ideas given in [5], [9], where a similar, model-theoretic approach was used for the semantic text analysis with logical inference. However, in contrast to our MRS-based approach, they apply *Discourse Representation Theory* [11] for the computation of full semantic representations. Furthermore, we use the framework *Heart of Gold* [7] as a basis for the semantic analysis. For a good overview of a combined application of deep and shallow NLP methods for RTE, we refer to [7], [12]. The application of logical inference techniques for RTE was already elaborately presented in [10], [13], [14]. A discussion on formal methods for the analysis of the meaning of natural language expressions can be found in [15].

III. System Architecture

Our system for RTE provides the user with a number of essential functionalities for syntactic, semantic, and logical textual analysis, which can selectively be overridden or specialized in order to provide new or more specific ones, e.g., for anaphora resolution or word sense disambiguation. In its initial form, the application supplies, among other things, flexible program interfaces and transformation components, allows for execution of a deep-shallow syntactic and semantic analysis, integrates external inference machines and background knowledge, maintains the semantic analysis and the inference process, and provides the user with a graphical interface for control and presentation purposes.

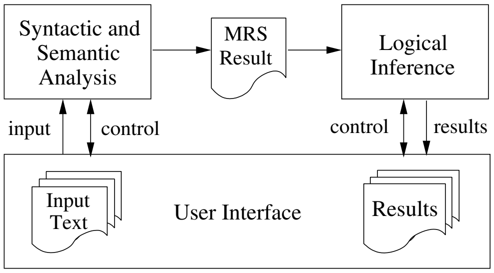


Figure 1. Overall architecture of the system.

In the following, we describe our system for RTE in more detail. It consists of three main modules (see Figure 1):

1) *Syntactic and Semantic Analysis*, where the combined deep-shallow semantic analysis of the input text is performed;
2) *Logical Inference*, where the logical inference process is implemented, supported by components with external knowledge and inference machines;
3) *Graphical User Interface*, where the analytical process is supervised and its results are presented to the user.

In the rest of the section, we discuss the way the particular modules of the system work. To make our description as comprehensible as possible, we make use of a small RTE problem. With its help we explain some crucial aspects of that how our system proceeds while trying to solve RTE problems. More specifically, we want to identify the logical relation between text T:

> *London's Tower Bridge is one of the most recognizable bridges in the world. Many falcons inhabit its old roof nowadays.*

and hypothesis H:

> *Birds live in London.*

To prove this textual entailment automatically, among other things, a precise semantic representation of the problem must be computed, the anaphoric reference between *Tower Bridge* and *its* in T must be resolved, and world knowledge (e.g., that *Tower Bridge* is in *London*) as well as ontological relations between the concepts (e.g., that *falcons* are *birds*) must be provided to the logical inference. We show how our system works while solving problems of such complexity.

A. Syntactic and Semantic Analysis

The texts of the input RTE problem after entering the system via the user interface go first through the syntactic processing and semantic construction of the first system module. To this end, they are analyzed by the components of the XML-based middleware architecture *Heart of Gold* (see Figure 2). It allows for a flexible integration of shallow and deep linguistics-based and semantics-oriented NLP components, and thus constitutes a sufficiently complex research instrument for experimenting with novel processing

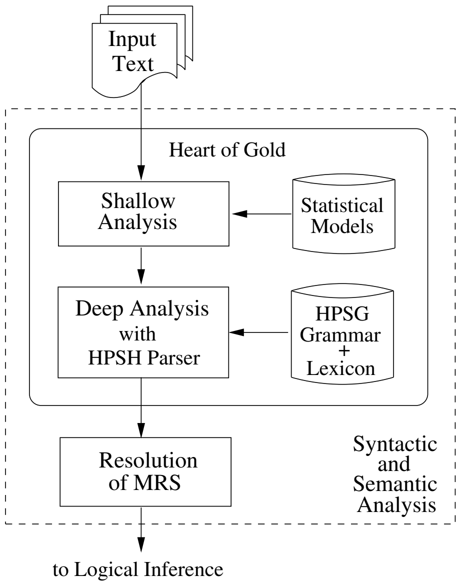


Figure 2. Module for syntactic and semantic analysis.

strategies. Here, we use its slightly modified standard configuration for English centered around the English Resource HPSG Grammar (ERG, see [16]). The shallow processing is performed through statistical or simple rule-based, typically finite-state methods, with sufficient precision and recall. The particular tasks are realized as follows: the tokenization with the Java tool JTok, the part-of-speech tagging with the statistical tagger TnT [17] trained for English on the Penn Treebank [18], and the named entity recognition with SProUT [19]. The latter one, by combining finite state and typed feature structure technology, plays an important role for the deep-shallow integration, i.e., it prepares the generic named entity lexical entries for the deep HPSG parser PET [20]. This makes sharing of linguistic knowledge among deep and shallow grammars natural and easy. PET is a highly efficient runtime parser for unification-based grammars and constitutes the core of the rule-based, fine-grained deep analysis. The integration of NLP components is done either by means of an XSLT-based transformation, or with the help of the *Robust Minimal Recursion Semantics* (RMRS, see [21]), when a given NLP component supports it natively. RMRS is a generalization of MRS. It can not only be underspecified for scope as MRS, but also partially specified, e.g., when some parts of the text cannot be resolved by a given NLP component. Thus, RMRS is well suited for representing output also from shallow NLP components. This can be seen as a clear advantage over approaches based strictly on some specified semantic representation like those presented, e.g., in [13], [22].

Furthermore, RMRS is a common semantic formalism for HPSG grammars within the context of the *LinGO Grammar Matrix* [23]. Besides ERG, which we use for English, there are also grammars for other languages like, e.g., the Japanese HPSG grammar *JaCY* [24], the *Korean Resource Grammar* [25], the *Spanish Resource Grammar* [26], or the proprietary German HPSG grammar [27]. Since all of those grammars can be used to generate semantic representations in form of RMRS, a replacement of ERG with another grammar in our system can be considered and thus a high degree of multilinguality achieved. To our best knowledge, it would be the first time that RTE problems in languages other than English could be considered.

The combined results of the deep-shallow analysis in RMRS form are transformed into MRS and resolved with Utool 3.1 [28]. Utool translates the input first from MRS into dominance constraints [29], a closely related scope underspecification formalism, and then enumerates in polynomial time all text readings represented by the dominance graph. In the current implementation, one of the most reasonable readings is chosen manually by the analyst for the further processing. A full automation of this task is still not possible in the current state-of-the-art. It requires much more knowledge about the RTE problem itself and about the discourse background. This important problem will be part of the further investigations.

For our small RTE example, the result of the combined syntactic and semantic analysis for H in form of RMRS, given as attribute value matrix, is presented in Figure 3. The results of the shallow analysis (marked bold) describe the named entities from H. Subsequently, the structure is transformed into MRS and resolved by Utool. The resulting first-order MRS in Prolog notation for the hypothesis H from our example is given below. The predicates with `_q_`, `_n_`, `_v_`, and `_p_` in their names represent quantifiers, nouns, verbs, and prepositions, respectively.

```
udef_q_rel(X6,
    bird_n_1_rel(X6),
    proper_q_rel( X9, and(
        named_rel(X9, london), and(
        locname_rel(london, X9),
        loctype_rel(city, X9))), and(
        live_v_1_rel(E2, X6),
        in_p_dir_rel(E10, E2, X9)))).
```

B. Logical Inference

The results of the semantic analysis in form of specified MRS combining deep-shallow predicates are translated into another, logical equivalent semantic representation FOLE (see Figure 4). The rule-based transformation conveys argument structure with a neo-Davidsonian analysis with semantic roles [30]. A definite article is translated according to the theory of definite description of Russell [31]. Temporal relations are modeled by adding additional predicates similar to [9], i.e., without explicit usage of time operators. Furthermore, it is possible to extend the translation mechanism to cover plural and modal forms. Appropriate ideas can be found, e.g., in [9], [32]. However, by applying them, one

$$\left[\begin{array}{ll}
\text{TEXT} & \\
\text{TOP} & h1 \\
\text{RELS} & \left\{\begin{array}{l}
\left[\begin{array}{ll} \textbf{locname\_rel} & \\ \text{LBL} & \mathit{h19000108} \\ \text{ARG0} & \mathit{x19000108} \\ \text{CARG} & \text{london} \\ \text{ARG1} & \mathit{x9} \end{array}\right]
\left[\begin{array}{ll} \textbf{loctype\_rel} & \\ \text{LBL} & \mathit{h19000110} \\ \text{ARG0} & \mathit{x19000110} \\ \text{CARG} & \text{city} \\ \text{ARG1} & \mathit{x9} \end{array}\right]
\left[\begin{array}{ll} \mathit{udef\_q\_rel} & \\ \text{LBL} & \mathit{h3} \\ \text{ARG0} & \mathit{x6} \\ \text{RSTR} & \mathit{h5} \\ \text{BODY} & \mathit{h4} \end{array}\right]
\left[\begin{array}{ll} \mathit{\_bird\_n} & \\ \text{LBL} & \mathit{h7} \\ \text{ARG0} & \mathit{x6} \end{array}\right] \\
\left[\begin{array}{ll} \mathit{\_live\_v} & \\ \text{LBL} & \mathit{h8} \\ \text{ARG0} & \mathit{e2} \\ \text{ARG1} & \mathit{x6} \end{array}\right]
\left[\begin{array}{ll} \mathit{\_in\_p} & \\ \text{LBL} & \mathit{h10001} \\ \text{ARG0} & \mathit{e10} \\ \text{ARG1} & \mathit{e2} \\ \text{ARG2} & \mathit{x9} \end{array}\right]
\left[\begin{array}{ll} \mathit{proper\_q\_rel} & \\ \text{LBL} & \mathit{h11} \\ \text{ARG0} & \mathit{x9} \\ \text{RSTR} & \mathit{h13} \\ \text{BODY} & \mathit{h12} \end{array}\right]
\left[\begin{array}{ll} \mathit{named\_rel} & \\ \text{LBL} & \mathit{h14} \\ \text{ARG0} & \mathit{x9} \\ \text{CARG} & \text{London} \end{array}\right]
\end{array}\right\} \\
\text{HCONS} & \{\mathit{h5} \text{ qeq } \mathit{h7}, \mathit{h13} \text{ qeq } \mathit{h14}\} \\
\text{ING} & \{\mathit{h8} \text{ ing } \mathit{h10001}\}
\end{array}\right]$$

Figure 3. RMRS as attribute value matrix for hypothesis H from the example.

needs to be careful since the complexity and the amount of the resulting FOLE formulas will grow rapidly, making the input problem apparently much harder to solve.

The translated FOLE formulas are stored locally and can be used for the further analysis. Furthermore, such formally expressed input text can and *should* be extended with additional knowledge in form of *background knowledge axioms*. The additional axioms are formulated in FOLE and integrated into the input problem. The integration of background knowledge will be discussed in detail in Section IV.

As an example here, the translation of the specified MRS into FOLE for the hypothesis H from our example given earlier in Section III-A produces the following formula with a neo-Davidsonian event representation:

```
some(X6,and(
    bird_n_1(X6),
    some(X9,and(and(
        named_r_1(X9),and(
        location_n_1(X9),and(
        london_loc_1(X9),
        city_n_1(X9)))),
        some(E2,and(
            event_n_1(E2),and(and(
            live_v_1(E2),
            agent_r_1(E2,X6)),
            in_r_1(E2,X9)))))))).
```

C. Inference Process

The goal here is to prove the logical relation between two input texts represented formally by corresponding FOLE formulas. We are interested in answering the question whether the relation is an entailment, a contradiction, or whether maybe the hypothesis H provides just new information with respect to the text T (i.e., is informative, see Section I). To check which type of a logical relation for the input problem holds, we use two kinds of automated reasoning tools:

- *Finite model builders*: Mace 2.2 [33], Paradox 3.0 [34], and Mace4 [35], and
- *First-order provers*: Bliksem 1.12 [36], Otter 3.3 [37], Vampire 8.1 [38], and Prover9 [39].

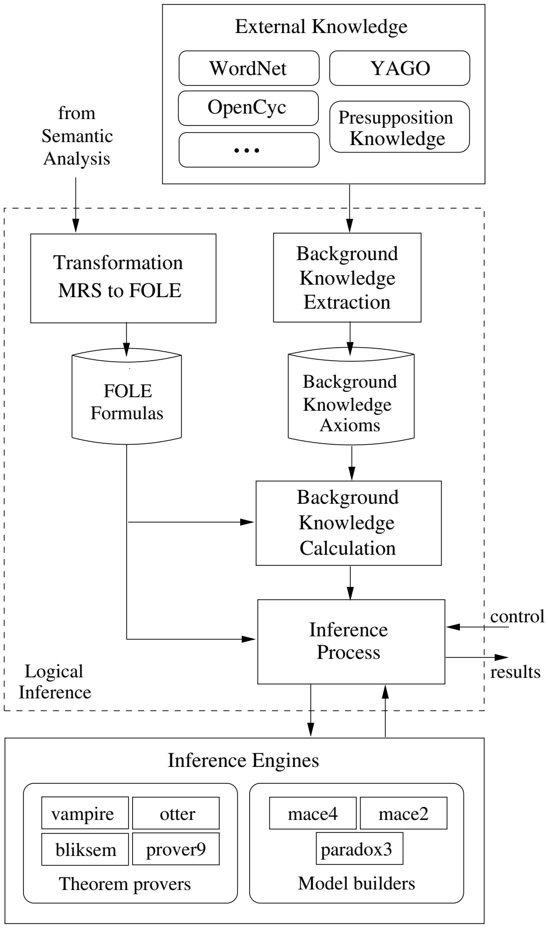


Figure 4. Module for logical inference with external inference machines and background knowledge.

While theorem provers are designed to prove that a formula is valid (i.e., the formula is true in any model), they are generally not good at deciding that a formula is not valid [40]. Model builders are designed to show that a formula is true in at least one model. The experiments with different inference machines show that solely relying on theorem proving is in most cases insufficient due to low recall. Indeed, our inference process incorporates model building as a central part of the inference process. Similar to [9], [40], we exploit the complementarity of model builders and theorem provers by applying them *in parallel* to the input RTE problem in order to tackle with its *undecidability* more efficiently. More specifically, the theorem prover attempts to prove the input whereas the model builder simultaneously tries to find a model for the negation of the input.

All reasoning machines were developed to deal with inference problems stated in FOLE. They are successfully integrated into our system for RTE. To this end, we use a translation from FOLE into the formats required by the inference tools. Furthermore, the user can specify via the user interface which inference machines (i.e., which theorem prover and which model builder) should be used by the inference process. The tests have shown that the efficiency and the success of solving a given RTE problem depend much on the inference machines chosen for it.

D. User Interface

The results of the syntactic processing, semantic construction, and logical inference like, e.g., HPSG and MRS structures, FOLE formulas, models, proofs, integrated background knowledge, and other detailed information are presented to the user within a dedicated GUI. With its help, one can further customize and control both the semantic and logical analysis, e.g., choose the input text or the background knowledge source, inspect the results of shallow-deep analysis, or select other inference machines.

IV. Improving the inference quality

Many applications in modern information technology utilize ontological background knowledge. This applies particularly to the applications from the Semantic Web, but also to other domains like, e.g., information retrieval, question answering, or recognizing textual entailment. The existing RTE applications today use typically only one source of background knowledge, e.g., WordNet [41] or Wikipedia. However, they could boost their performance if a huge ontology with knowledge from several sources were available. We show here how more than one knowledge source can be used successfully for RTE. In this paper, we mean by ontology any set of facts and/or axioms comprising potentially both individuals (e.g., London) and concepts (e.g., city).

The inference process needs background knowledge to support its proofs. However, with increasing number of background knowledge axioms the search for finite models becomes more time-consuming. Thus, only problem-relevant knowledge should be considered in the inference process.

A. Background Knowledge

Our RTE system supports the extraction of background knowledge from different kinds of sources (see Figure 4). It supplies problem-relevant background knowledge automatically as first-order axioms and integrates them into the input RTE problem. WordNet 3.0 is used as lexical knowledge source for synonymy, hyperonymy, and hyponymy relations. With WordNet we try to detect entailments between lexical units from the text and the hypothesis. Axioms of generic knowledge cover the semantics of possessives, active-passive alternation, and spatial knowledge (e.g., that Tower Bridge is located in London). YAGO [42] with facts automatically extracted from Wikipedia and unified with WordNet is used as a source of ontological knowledge. OpenCyc 2.0 [43] can also be used as a background knowledge source. The computation of axioms for a given problem is solved using a variant of Lesk's WSD algorithm [44].

In the following, we describe the idea we use to combine individuals and concepts from WordNet with those from YAGO in order to support RTE. Our integration technique is composed of two steps. After the first-order representation of the problem is computed and subsequently translated into FOLE, the search for relevant background knowledge begins. First, we list all predicates (i.e., concepts and individuals) from the FOLE formulas which can be used for the search. In the current implementation, we consider as *search predicates* S all nouns, verbs, and named entities, together with their sense information (i.e., their readings) specified by the last number in the predicate name, e.g., `bird_n_1`. Having the search predicates, we try to find them in WordNet and, by employing the hyperonymy/hyponymy relation, we build a *knowledge tree* T_K with leaves represented by the concepts from the formulas, whereas inner nodes and the root are coming from WordNet.

In Figure 5, we show a fragment of a knowledge tree for $\{T, H\}$ of our RTE problem from the beginning of Section III. Here, each node represents at least one concept or individual, whereas the directed edges correspond to the hyponym relations between them, e.g., the named entity `london` is a hyponym of the concept `city`. Note that in the opposite direction they describe the hyperonym relations, e.g., the concept `city` is a hyperonym of the named entity `london`. Figure 5 depicts also one complex node representing synonymous concepts `live` and `inhabit`.

It is crucial for the integration that the sense information computed for the concepts and individuals during the semantic analysis matches exactly the senses used by external knowledge sources. This ensures that the semantic consistency is preserved across the semantic and logical analysis. However, this constitutes an extremely difficult task which does not seem to be solved fully automatically yet by any word sense disambiguation technique. Since in WordNet but also in ERG the senses are ordered from most to least frequently used, with the most common sense numbered `1`, we take in the current implementation for semantic representations generated during the semantic analysis the most frequent concepts from ERG.

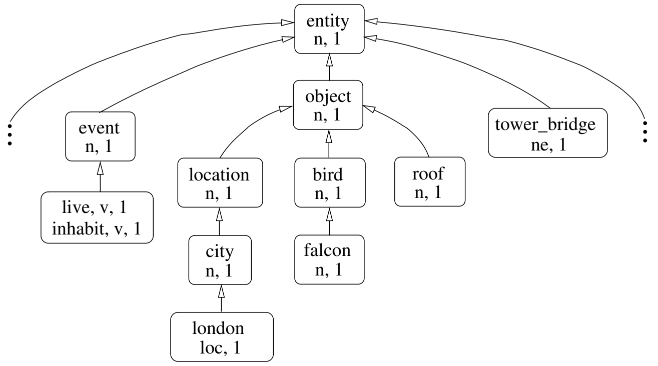


Figure 5. Example of knowledge tree for RTE. Here, v, n, loc, and ne stand for verb, noun, location, and named entity, respectively, whereas the numbers represent the sense information.

In the second step of our integration technique, we consult YAGO about the predicates from S that were not found in WordNet during the first step. If succeed, YAGO returns a directed acyclic graph (DAG) G_K with new concepts which classify those concepts that were not recognized before. Unfortunately, as a DAG, it cannot be integrated completely into the knowledge tree T_K. Our experiments have shown that a knowledge graph, when represented as a tree, assures that the set of background knowledge axioms, which will be generated afterwards from that tree stays consistent (i.e., it includes no contradictions). Thus, in order to preserve the consistency and correctness of the results, we select for the integration into the knowledge tree T_K only those concepts and relations from G_K, which lay on the longest path from the root to one of its leaves and which has the most common nodes with the knowledge tree T_K from the first step. This heuristic can cause some loss of effectivity of the entire RTE inference process, since some concepts which are relevant for the RTE problem might not be integrated as background knowledge into it. Nevertheless, because of its acceptable performance while solving problems from the development sets of the past RTE Challenge [4], we have decided to use it as a good starting point for the further research.

After the background knowledge tree T_K has been extended, the knowledge axioms are generated from it. We generate axioms expressing the hyperonymy/hyponymy relations (i.e., ontological relations is-a and is-not-a) and the synonymy relations (is-eq) in T_K. For the knowledge tree given in Figure 5, the following axioms (here not a complete list) can be generated.

$$\begin{gathered}\forall x(city\_n\_1(x) \rightarrow location\_n\_1(x))\\ \forall x(event\_n\_1(x) \rightarrow \neg object\_n\_1(x))\\ \forall x(live\_v\_1(x) \leftrightarrow inhabit\_v\_1(x))\end{gathered}$$

B. Presupposition resolution

Many words and phrases trigger presuppositions which have clearly semantic content important for the inference process. We try to represent some of them explicitly. Our trigger-based mechanism uses noun phrases as triggers, but it can be extended to verb phrases, particles, etc. After a presupposition is triggered, the mechanism resolves it, and integrates it as a new FOLE axiom into the RTE problem. The automatic axiom generation is based on λ-conversion and employs *abstract axioms* and a set with possible *axiom arguments*. The axioms and their arguments are still part of an experimental knowledge source (see Presupposition Knowledge in Figure 4). Here is an example for an abstract axiom which allows for a translation from a noun phrase into an intransitive verb phrase:

$$\begin{aligned}&\lambda P[\lambda R[\lambda S[\forall x_1(\forall x_2(P@x_1 \wedge R@x_2 \wedge nn\_r\_1(x_1, x_2)\\ &\rightarrow \exists x_3(R@x_3 \wedge \exists x_4(S@x_4 \wedge event\_n\_1(x_4) \qquad (1)\\ &\quad \wedge agent\_r\_1(x_4, x_3)))))]]].\end{aligned}$$

If text T (expressed with FOLE formulas) contains a noun phrase being a key for some entry in the set of possible axiom arguments, then the arguments pointed by that key are applied to their abstract axiom, and a new background axiom is generated. For a complex noun phrase *price explosion* with its semantic representation $price\_explosion\_n\_1$, the following arguments can be considered:

$$\lambda x[explosion\_n\_1(x)],\ \lambda x[price\_n\_1(x)],\ \text{and}\\ \lambda x[explode\_v\_1(x)],$$

which after being applied to the abstract axiom (1) produce the following background knowledge axiom:

$$\begin{aligned}&\forall x_1(\forall x_2(explosion\_n\_1(x_1) \wedge price\_n\_1(x_2) \wedge\\ &nn\_r\_1(x_1, x_2) \rightarrow \exists x_3(price\_n\_1(x_3) \wedge\\ &\exists x_4(explode\_v\_1(x_4) \wedge event\_n\_1(x_4) \wedge\\ &agent\_r\_1(x_4, x_3))))). \qquad (2)\end{aligned}$$

The presupposition axioms having complexity similar to (2) are first combined with the existing background knowledge axioms and finally integrated as background knowledge into the input RTE problem.

V. Conclusion and Future Work

In this paper, a new adaptable, linguistically motivated system for RTE was presented. Its deep-shallow semantic analysis, employing a broad-coverage HPSG grammar ERG, was combined with a logical inference process supported by an extended usage of external background knowledge. The architecture of the system was given in detail and its functionality was explained with several examples.

The system was successfully implemented and evaluated in terms of success rate and efficiency. For now, it is still impossible to measure its exact semantic accuracy as there is no corpus with gold standard representations which would make comparison possible. Measuring semantic adequacy could be done systematically by running the system on controlled inference tasks for selected semantic phenomena.

For our tests, we used the RTE problems from the development set of the third RTE Challenge [4]. Our system with was able to solve correctly 64 percent of the RTE problems. This is better than the most of the other approaches from that RTE Challenge which are based on some deep approach combined with logical inference. Unfortunately, it is still not as good as the success rate of 72 percent obtained by the best logic-based semantic approach given by [14]. This can be explained, among other things, by a more extensive and fine-grained usage of specific semantic phenomena, e.g., a sophisticated analysis of named entities, in particular person names, distinguishing first names from last names. This shows, however, that extending our system with similar techniques for more accurate treatment of specific semantic phenomena should further improve its success rate.

Nevertheless, it is interesting to look at the inconsistent cases of the inference process which were produced during the evaluation. They were caused by errors in presupposition and anaphora resolution, incorrect syntactic derivations, and inadequate semantic representations. They give us good indications for further improvements. Here, particularly the word sense disambiguation problem will play a decisive role for matching the set of senses of the semantic analyzers with multiple, and likely different, sets of senses from the different knowledge resources. Once tackled more precisely, it should decisively improve the success rate of the system.

As being still work-in-progress, we plan to extend our system with methods for word sense disambiguation, paraphrase detection, and a better anaphora resolution within a discourse. We consider also enhancing the logical inference module with statistical inference techniques in order to improve its performance and recall. Since the strength but in some respects also the weakness of our system lies in the difficulties regarding the computation of a full semantic representation of the input problem (see, e.g., [45] for a good discussion), it might be recommended to integrate into the system some models of natural language inference which identifies valid entailments by their lexical and syntactic features, without full semantic interpretation like, e.g., the one proposed by MacCartney and Manning [46].

ACKNOWLEDGMENT

The authors would like to thank Matthias Hecking for many fruitful discussions and comments on the work presented in this paper.

REFERENCES

[1] I. Dagan, B. Dolan, B. Magnini, and D. Roth, “Recognizing textual entailment: Rational, evaluation and approaches,” *Natural Language Engineering. Special Issue on Textual Entailment*, vol. 15, no. 4, pp. i–xvii, 2009.

[2] L. Bentivogli, I. Dagan, H. T. Dang, D. Giampiccolo, and B. Magnini, “The fifth PASCAL recognizing textual entailment challenge,” in *TAC 2009 Workshop*, Gaithersburg, Maryland, 2009.

[3] J. Bos, “Towards wide-coverage semantic interpretation,” in *Proceedings of the 6th International Workshop on Computational Semantics IWCS-6*, 2005, pp. 42–53.

[4] D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan, “The third PASCAL recognizing textual entailment challenge,” in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic, 2007, pp. 1–9.

[5] P. Blackburn and J. Bos, *Representation and Inference for Natural Language. A First Course in Computational Semantics*. CSLI, 2005.

[6] E. Akhmatova, “Textual entailment resolution via atomic propositions,” in *Proceedings of the First PASCAL Challenges Workshop on Recognising Textual Entailment*, Southampton, UK, 2005, pp. 61–64.

[7] U. Schäfer, “Integrating deep and shallow natural language processing components – representations and hybrid architectures,” Ph.D. dissertation, Saarland University, Saarbrücken, Germany, 2007.

[8] A. Copestake, D. Flickinger, C. Pollard, and I. A. Sag, “Minimal recursion semantics: An introduction,” *Research on Language and Computation*, vol. 3, pp. 281–332, 2005.

[9] J. R. Curran, S. Clark, and J. Bos, “Linguistically motivated large-scale NLP with C&C and boxer,” in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, Prague, Czech Republic, 2007, pp. 33–36.

[10] J. Bos and K. Markert, “When logical inference helps determining textual entailment (and when it doesn’t),” in *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*, Venice, Italy, 2006.

[11] H. Kamp and U. Reyle, *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: Kluwer Academic Publishers, 1993.

[12] J. Bos and K. Markert, “Combining shallow and deep NLP methods for recognizing textual entailment,” in *Proceedings of the First PASCAL Challenges Workshop on Recognising Textual Entailment*, Southampton, UK, 2005, pp. 65–68.

[13] ——, “Recognising textual entailment with logical inference,” in *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Vancouver, Canada, 2005, pp. 628–635.

[14] M. Tatu and D. Moldovan, “A logic-based semantic approach to recognizing textual entailment.” in *Proceedings of the COLING/ACL on Main conference poster sessions*, Morristown, NJ, 2006, pp. 819–826.

[15] J. Bos, “Let’s not argue about semantics,” in *Proceedings of the Sixth International Language Resources and Evaluation (LREC’08)*, Marrakech, Morocco, 2008, pp. 28–30.

[16] D. Flickinger, “On building a more efficient grammar by exploiting types,” *Natural Language Engineering*, vol. 6, no. 1, pp. 15–28, 2000.

[17] T. Brants, “TnT – a statistical part-of-speech tagger,” in *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000*, Seattle, WA, 2000, pp. 224–231.

[18] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, “Building a large annotated corpus of English: The Penn Treebank,” *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[19] W. Drożdżyński, H.-U. Krieger, J. Piskorski, U. Schäfer, and F. Xu, “Shallow processing with unification and typed feature structures – foundations and applications,” *Künstliche Intelligenz*, vol. 18, no. 1, pp. 17–23, 2004.

[20] U. Callmeier, “PET – a platform for experimentation with efficient HPSG processing techniques,” *Natural Language Engineering*, vol. 6, no. 1, pp. 99–108, 2000.

[21] A. Copestake, “Report on the design of RMRS,” University of Cambridge, UK, Tech. Rep. D1.1b, 2003.

[22] P. Blackburn, J. Bos, M. Kohlhase, and H. D. Nivelle, “Automated theorem proving for natural language understanding,” in *Problemsolving Methodologies with Automated Deduction (Workshop at CADE-15)*, 1998.

[23] E. M. Bender, D. Flickinger, and S. Oepen, “The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars,” in *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, 2002.

[24] M. Siegel and E. M. Bender, “Efficient deep processing of japanese,” in *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization. Coling 2002 Post-Conference Workshop*, 2002.

[25] K. Jong-Bok and Y. Jaehyung, “Parsing mixed constructions in a typed feature structure grammar,” *Lecture Notes in Artificial Intelligence*, vol. 3248, pp. 42–51, 2005.

[26] M. Marimon, “Integrating shallow linguistic processing into a unification-based spanish grammar,” in *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, 2002.

[27] B. Cramer and Y. Zhang, “Construction of a German HPSG grammar from a detailed treebank,” in *Proceedings of the Workshop on Grammar Engineering Across Frameworks*. Association for Computational Linguistics, 2009.

[28] A. Koller and S. Thater, “Efficient solving and exploration of scope ambiguities,” in *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, Ann Arbor, Michigan, 2005, pp. 9–12.

[29] S. Thater, “Minimal recursion semantics as dominance constraints: Graph-theoretic foundation and application to grammar engineering,” Ph.D. dissertation, Saarland University, Saarbrücken, Germany, 2007.

[30] D. Dowty, “On semantic content of the notion of ”thematic role”,” in *Properties, Types and Meaning*, G. C. Barbara Partee and R. Turner, Eds. Dordrecht (Kluwer), 1989, vol. 2, pp. 69–129.

[31] B. Russell, “On denoting,” *Mind, New Series*, vol. 14, no. 56, pp. 479–493, 1905.

[32] H. Lohnstein, *Formale Semantik und natürliche Sprache. Einführendes Lehrbuch.* Westdeutscher Verlag, 1996.

[33] W. McCune, *Mace 2.0 Reference Manual and Guide*, Argonne National Laboratory, IL, 2001.

[34] K. Claessen and N. Sörensson, “New techniques that improve MACE-style model finding,” in *Proceedings of the CADE-19 Workshop: Model Computation – Principles, Algorithms, Applications*, Miami, FL, 2003.

[35] W. McCune, *Mace4 Reference Manual and Guide*, Argonne National Laboratory, IL, 2003.

[36] H. de Nivelle, “Bliksem 1.10 user manual,” URL: http://www.ii.uni.wroc.pl/˜nivelle/software/bliksem, 2003.

[37] W. McCune, *OTTER 3.3 Reference Manual*, Tech. Memo ANL/MCS-TM-263, Argonne National Laboratory, Argonne,IL, Argonne National Laboratory, IL, 2003.

[38] A. Riazanov and A. Voronkov, “The design and implementation of VAMPIRE,” *AI Commun.*, vol. 15, no. 2,3, pp. 91–110, 2002.

[39] W. McCune, “Prover9 manual,” URL: http://www.cs.unm.edu/˜mccune/prover9/manual/2009-11A/, Argonne National Laboratory, Argonne, IL, 2009.

[40] J. Bos, “Exploring model building for natural language understanding,” in *Proceedings of ICoS-4*, 2003, pp. 25–26.

[41] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, MA, 1998.

[42] F. Suchanek, G. Kasneci, and G. Weikum, “YAGO - a large ontology from Wikipedia and WordNet,” *Elsevier Journal of Web Semantics*, vol. 6, no. 3, pp. 203–217, 2008.

[43] 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*, Stanford, CA, 2006.

[44] S. Banerjee and T. Pedersen, “An adapted Lesk algorithm for word sense disambiguation using WordNet,” in *Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Text Processing*, London, UK, 2002, pp. 136–145.

[45] A. Burchardt, N. Reiter, S. Thater, and A. Frank, “A semantic approach to textual entailment: System evaluation and task analysis,” in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, 2007.

[46] B. MacCartney and C. D. Manning, “An extended model of natural logic,” in *Proceedings of the 8th International Conference on Computational Semantics (IWCS-8)*, 2009, pp. 140–156.

Enriching Videos with Light Semantics

Smitashree Choudhury
Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland
smitashree.choudhury@deri.org

John G. Breslin
School of Engineering and Informatics
National University of Ireland, Galway
Galway, Ireland
john.breslin@nuigalway.ie

Abstract—**This paper describes an ongoing prototypical framework to annotate and retrieve web videos with light semantics. The proposed framework reuses many existing vocabularies along with a video model. The knowledge is captured from three different information spaces (media content, context, document). We also describe ways to extract the semantic content descriptions from the existing user-generated content using multiple approaches of linguistic processing and Named Entity Recognition, which are later identified with DBpedia resources to establish meanings for the tags. Finally, the implemented prototype is described with multiple search interfaces and retrieval processes. Evaluation on semantic enrichment shows a considerable (50% of videos) improvement in content description.**

Keywords - social media; multimedia semantics; semantic web; linked open data; semantic search

I. Introduction

With the huge increase of user videos on the Web, the traditional search paradigm is proving to be ineffective in discovering and browsing interesting videos. Moreover, due to the complex nature of multimedia, reusability of video documents is very low, and as a result, almost every time a user has to create their video from scratch. We need better mechanisms to organise and represent the video data in order to address the above issues. Meaningful organisation and metadata representation is one of the concerns, but is as yet largely overlooked for multimedia. At present, user videos may come with certain embedded metadata, either created by users while publishing or during the production workflow, such as camera settings (though these are still not easily accessible in the case of web video). Some of the useful meta information is also created in the course of usage and sharing amongst users after publishing. Information such as free labels as tags, descriptions, user responses to the video, location information, membership in various groups, captions inside the video are immensely useful. The problem with the existing situation is that even if we collect and process this information, reusability (the data integration problem) remains elusive because of the lack of any formal semantics attached to the videos. Tags are freeform words with implicit meaning and relations known to the creator or publisher. The problems of user tagging have been explored well in many research studies. The major challenges are as follows. (1) Tag variation: different tags are used for the same kind of resources, e.g., “New York City”, “NYC”. There is no explicit way to express that these two tags are indeed meant to be the same. (2) Polysemy tags: a single tag used for different meanings. This problem occurs due to a difference in understanding of a user about the resource he or she is tagging, and may also depend on sociocultural differences among users. (3) Lack of formal structure among tags makes it difficult to understand, classify and recommend tags automatically. Besides these issues, we have problems with misspelling, compound tags such as “globalwarming”, multiword tags expressed as multiple tags, and tags used out of a community consensus such as “SEMAPRO2010”.

This plethora of information can be harnessed to add an extra layer of machine-readable metadata that will help to understand the opaque media data a little better. There are many well-defined and comprehensive formal ontologies available to describe media structures and content. The earliest such effort was made by the MPEG (Motion Picture Expert Group) community in developing MPEG-7 [7], a standard for describing media, but it failed to take hold significantly due to its lack of formal semantics and interoperability issues. The Semantic Web community made efforts [5] to convert MPEG-7 to RDFS (Resource Description Framework Schema) representations, in order to avail of the benefits offered by Semantic Web technologies such as RDF (the Resource Description Framework). However, the complexities of MPEG-7 prevented it from being fully converted and many data type issues remain unresolved. Media ontologies such as COMM [4] took a pure Semantic Web approach to describe and represent media with its different granularities. Many ontologies were developed to address domain-specific media such as museum collections, the football domain, etc.

Recently, the W3C Media Annotation Working Group has made an effort to devise a comprehensive media ontology to describe video on the Web, which may become a recommended standard in the near future. In spite of many concerted efforts, it is hard to see any widespread usage of these vocabularies. The reasons are not well studied, but on the other side we can see that there are some vocabularies such as FOAF (Friend of a Friend) [14], [20], SIOC (Semantically-Interlinked Online Communities) [13], which have been adopted quite well and quickly. We assume that the reasons for such adaptability may be due to their inherent simplicity and easy-to-understand characteristics. Keeping in mind the above challenges, we adopted the principle of keeping it short and simple (KISS), yet fulfilling the basic requirements of ontology engineering, and proposed a lightweight framework to describe web videos. The approach makes use of many existing vocabularies such as Dublin Core, FOAF and SIOC wherever possible along with our own model. In spite of a very small and light framework, it covers almost every aspect of a media description. The description is broadly categorised under three sub modules: (a) document and media properties; (b) semantic content description; and (3) social context descriptions. Fig. 1 shows a subset of attributes from each of the three contributing information spaces. The details of the proposed model are in [12]. One of the focal points of the framework is its easy computability in the sense that most of the classes can be automatically populated with instances with little processing rules and heuristics. We have kept in mind the fact that in the future we may have to devise ways to map with the standard media ontology recommended by the W3C.

We also aim to link identified concepts to those of the Linked Open Data initiative (LOD), which was started in 2007 with the objective of creating a Web of Data connected to each other following four basic principles [11]. The hub of the Linked Open Data cloud is DBpedia, which is the RDF representation of Wikipedia [22] articles, categories and info boxes. Wikipedia is the largest user-generated multi-lingual encyclopedia in the world, maintained by tens of thousands of users since 2001. Other domain specific data sources such as book data, scientific publication data, life science data, geographical data are all connected to DBpedia [26] in the cloud. The present size of the LOD is more than 8 billion triples and is constantly increasing in size. More details of the LOD initiative can be found in [11].

The rest of this paper is structured as follows: in Section 2, we describe the related work. Section 3 describes the implementation flow including modeling, populating the model integration with linked data. Section 4 shows our semantic search prototype. Section 5 concludes this paper.

II. Related Studies

This section describes various studies related to semantic media modeling and semantic search of media data focusing on video search. It will also describe some efforts towards ontology learning from folksonomies. Ontology learning from folksonomies follows different approaches. Researchers in [6] suggested lightweight ontology learning from a folksonomy based on broader and narrower semantic relations. Passant [8] exploited folksonomies to populate a corporate ontology. Specia and Motta [10] used methods to cluster similar tags and find a match in an existing ontology.

Other studies proposed data mining technologies to mine the structural information from user tags. Schmitz et al. [9] used association rule mining techniques to recommend tags. Regarding semantic search, not much work has been carried out in the domain of multimedia data. A comprehensive study of semantic search is described in [1] while [2] describes an ontology-based search engine. A semantic video search system is described in [18]. Swoogle [17] and Sindice [3] are two major search engines focused on existing Semantic Web data.

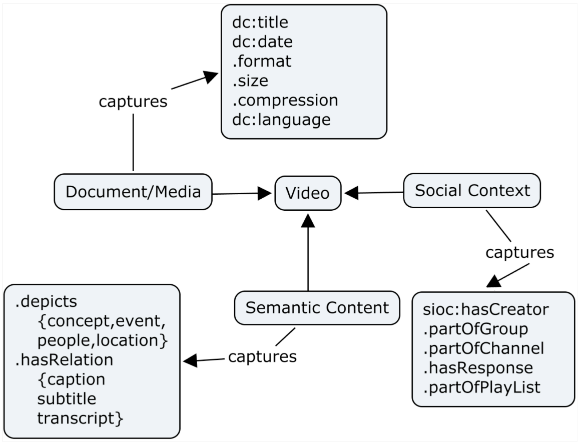


Figure 1. A subset of the video model.

III. Implementation Architecture

This section describes various aspects of the prototypes including the instance creation, video annotation and retrieval modules. Fig. 2 shows the architectural flow of the prototype.

A. Data Collection

We used APIs and RSS feeds for different video sharing sites such as YouTube [23] and Vimeo [24] to collect the video metadata. Metadata includes title, description, tags, date, number of views, ratings, groups, duration, location data, etc. We have collected 10,000 video items for the prototype in the domain of science and technology.

B. Modelling Web Video

Our model for video description (Fig. 1) covers three major areas such as video document and media properties, social context attributes and depicted semantic content. The above proposed modeling approach not only satisfies the general ontological requirements such as modularity, interoperability and extensibility, but also separation of concern specifically aimed for media semantics. The uniqueness of the proposed approach for describing video is its simplicity and ease of use. Regarding the document level description, it is a widely-accepted practice to use Dublin Core terms such as title (*dc:title*) and creation date (*dc:date*), but media documents also carry some media-specific technical attributes such as format (*sva:format*), duration (*sva:duration*), etc., which are described using the video model described in [12]. Regarding the content description, video content can be described with different granularities starting with a global description (*dc:description*) to segments created by temporal and spatial decomposition. Segment content can be captured through the *sva:depicts* attribute whose range may be topic, event, geo-location, *foaf:Person* or *skos:Concept* as per requirements. The recent growth of social media interaction on the Web has made all objects on the Web somewhat social, thus we can embed some emerging properties such as comments, ratings, group membership, etc. For describing social contextual properties, the best-suited vocabulary is SIOC ontology. Its goal is to describe objects and interactions in online communities. We consider the publisher of a video as an instance of *sioc:UserAccount* which belongs to a *foaf:Peson*. Video is an item in a *sva:Channel* which is a subclass of *sioc:Container*.

C. Content Processing for Concept Learning

Any ontology-based knowledgebase requires the instances to be populated manually, semi-automatically or by automatic means. Since manual annotation is not feasible and scalable, we tried to accomplish this semi-automatically by exploiting the existing information and getting user feedback in case of higher uncertainties such as the absence of any user data. APIs and RSS feeds offer an easy-to-go solution for many of the document level properties such as title, description, duration, categories, etc. which can be directly transformed to the Dublin Core properties or other global properties, but the real challenges come while creating the content description instances. The user-generated content is free text, devoid of any formal structure. In order to achieve the implicit formal structure, the content needed to be processed and normalised with various approaches before being mapped to any kind of ontological concepts.

Pre-processing of textual data involves:

- removing stop words
- removing tags with less than two characters
- removing username tags

After basic pre-processing we followed a few more intensive cleaning tasks in order to get some sensible tags from the data.

Multi-Term Tags: Tags with multiple words are one of the other major problems while identifying semantic entities. Mostly users enter multiple words as part of a single tag, and each of the tags are supposed to be separated by a comma delimiter, but the API gives a single word as a single tag. Taking the same example used previously, in many cases the YouTube API gives "global" and "warming" as two different tags while a single tag of "global warming" is more descriptive and accurate. In order to clean the tag space further and in the hope of getting some phrase tags, we followed a few simple syntactic rules (shown below) to parse the tag space. Examples of such rules are widely used in natural language processing research. After identifying the patterns, we check the resulting phrase with Wikipedia concepts, and if a match is found we keep the phrase as a possible candidate for a tag.

((Noun)+(Noun)*) or (Noun-Prep)?+(Adj|Noun)*

TABLE I. EXAMPLE OF MULTI TERM TAG IDENTIFICATION

Original tag space	Identified multi-term tags
sequencing, dna, rna, sanger, gilbert, big, dyes, terminators, molecular, biology, genomics, secuenciacin, adn, cidos, nucleicos	sequencing, dna, rna, sanger, gilbert, big, dyes, terminators, molecular biology, genomics

Entity Recognition (NER) with Open Calais: Open Calais [27] is a free non-commercial web service from Thomson Reuters for identifying various semantic entities such as person, event, location, company, dates, organisations, concepts, etc. Though its application is aimed at well-formed textual documents, we have tried it on tag spaces and description content as an experiment. The effectiveness of NER in tag spaces is expected to be lower because tags are independent words without any syntactic structure and grammar rules, but we assume that with careful cleaning and normalisation, we may be able to identify some entities. At present, entities identified from the tag space are only accepted if they are supported from other sources. When the video has more description content, use of Open Calais improves the result. Table II below shows five different identified entities from a video description.

TABLE II. EXAMPLE OF ENTITY IDENTIFICATION

Description content	Identified entities
Thus far, most DNA sequencing has been performed using the chain termination method developed by Frederick Sanger. This technique was also used to sequence the genome of James Watson recently. Pathogens may lead to treatments for contagious diseases. Biotechnology is a burgeoning discipline…	Contagious diseases Frederick Sanger (Person) James Watson (Person) Biotechnology (tech) DNA sequencing (tech)

Compound Tags: Users create tags with no white space, e.g., "globalwarming" which is a concatenation of two words "global" and warming". These tags are useful, but not in their original form, so we need to process them in order to separate the words with a whitespace and form a proper tag. We followed a few simple heuristics to identify meaningful words from a tag. The pseudo code is given below.

```
• Divide the tag ($T_i$) into two sub tags ($t_1$, $t_2$) where length of $t_1$ is length(($T_i$)/2)+1 and $t_2$ is length(($T_i$)/2)-1
• Check if $t_1$ exists in the dictionary
• If($t_1$ exists) = true
      o Check if $t_2$ exists
      o Form tag with $t_1$+WS+$t_2$ (equation 1)
• Else
      o Offset $t_1$ or $t_2$ with one character and check
      o If (one exists) then concatenate the offset and check if the other exists
            ▪ Form the tag with $t_1$+WS+$t_2$ (equation 2)
      o Else (follow equation 3)
```

Equation 3

If equation 2 fails, then we divide and create a third term t_3 with the offset characters and check iteratively. When two are found in the dictionary, we add the third by default and form the tag by adding a WS in between the terms. Though this is a brute force method it gives a satisfactory result for improving the tag quality

We restricted compound tags to a maximum of three terms. An example of the above algorithm is given below in Table III.

TABLE III. EXAMPLE OF COMPOUND TAG DECOMPOSITION

Original tag ("globalwarming")
Step 1. globalw (= t_1) and arming (= t_2)
Step 2. If (globalw is present in dictionary) = no
Step 3. Offset by 1 from t_1 (globalw-w = global) and add to t_2 (w+arming=warming)
Step 4. Check if t_1 and t_2 exists in dictionary = yes
Step 5. Form tag $T_i = t_1$+WS+t_2= **global warming**

D. Integrating with the Linked Open Data (LOD) Cloud

A video can be interlinked with multiple data sources such as geographical data, a *foaf:Person* or DBpedia concepts. Instances of concept, person, event, location are mapped with the property *owl:sameAs* or *rdfs:seeAlso*.

The focus here will be on content linking, from a tag to a Wikipedia concept to a DBpedia resource, e.g., the tag "E.coli" is mapped to a Wikipedia concept "Escherichia coli" and subsequently to the DBpedia resource "http://dbpedia.org/resource/Escherichia_coli". DBpedia is the hub of the LOD cloud, so any mapping to DBpedia will ultimately lead to other domain-specific data such as life science data or movie data.

Since there may not always be a one-to-one mapping between a user tag and an ontological concept, we need some kind of entity resolution mechanism. Here we computed a similarity between user tags and wiki concepts (from wiki articles) and redirect concepts, and derive the top match as the identified concept. This particular similarity is computed with a Lucene index of Wikipedia articles, redirects and categories.

E. Semantic Relation Extraction

Once we get a list of probable tags from all of the above steps, we need to formally ground them with some ontological concepts with relations between them. Since at all stages in the above processing we verified the possible tag against an index of Wikipedia articles, categories and redirect concepts, they are more or less considered ontological concepts though the relationship between them is still unclear and vague.

To extract the relationship between tags we need to compute the similarity between tags. Many studies explored tag similarity using various approaches and distributional measures such as co-occurrence similarity [16], Folkrank [15], etc. At the time of writing, this similarity module has not been implemented, but we plan to exploit the link structure of Wikipedia articles to estimate the semantic distance between tags.

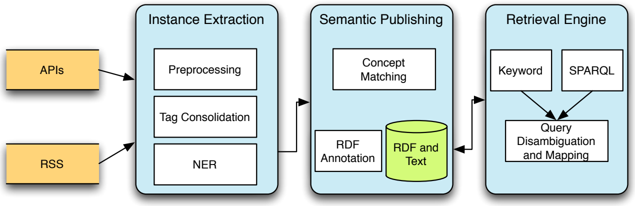


Figure 2. System architecture modules.

IV. SEARCH MODULE

Machine-readable data will facilitate complex query answering which was not possible before. It will also help to infer some unseen relations existing between various data pieces within the knowledgebase (KB) itself, but it still remains insulated from the huge amount of data lying outside the KB which may hold much more relevant and useful information both known and unknown.

Here come the benefits of linked data: by following some simple principles we can make our data accessible to other datasets and vice versa. The benefits of linked data can only be realised with practical applications, so we have decided to enable our semantic search module to explore the linked data to facilitate navigational search, where the user can explore and discover much related information and therefore reformulate their queries. Fig. 3 shows an interface for the query "Albert Einstein", and its related information as aggregated from the DBpedia source.

A. User Interface

The role of good user interfaces for Semantic Web data has largely been overlooked. To our understanding, it is one of the major contributing factors to the slow adoption of Semantic Web technologies. Although recently some efforts have been made to address the issue, such as faceted browsers like mSpace [19] and Sigma [21], the problem is far from over.

The ideal solution should not reflect underlying data complexities but still give the benefits of semantic search. [22] is a standard recommendation for querying Semantic Web data, but exposing a SPARQL interface as the primary query interface will be riskier as learning a complex query language will hardly be welcomed by users other than concerned geeks. A simple keyword-based interface may suffice for most users, but will lose the complex query answering mechanisms possible with semantic data.

Therefore we have planned to expose different levels for a query interface in order to facilitate complex queries by exposing underlying data properties with each querying stage. We move from keyword search to faceted search, where the major facets are dynamically constrained for each iteration, and finally to navigational search. Navigational search enables the user to access an integrated view of the query term. Fig. 3 shows the incremental query interface of the system. The first point of entry is a dual interface of keyword search and SPARQL end point. The result of the first query is deployed in a faceted interface. Details of the video are exposed in a navigational space where related facts are connected DBpedia resources.

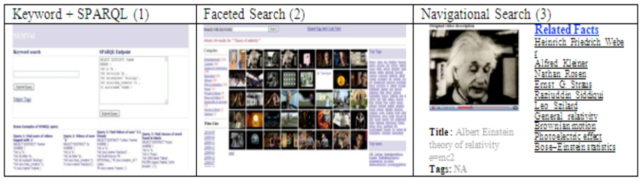


Figure 3. Three different interfaces.

B. Retrieval Engine Architecture

Since semantic search is defined as the matching between query semantics and content semantics, we need to capture the query semantics before the actual search process. User query intention can be captured in different ways starting from interacting with query reformulation to automatic disambiguation of a query.

- For the keyword search interface, we have adopted a simple approach to disambiguate the user queries by mapping the query term(s) to the best possible semantic entity that exists in the knowledgebase. In the case of more than one semantic entity, entity resolution is performed in favour of the most popular one, followed by the rest. However, in such cases, precision goes down. We need to adopt a more robust entity resolution mechanism in order to improve the search quality.
- At the second stage, the query is sent to the Lucene index for retrieval. The results are clustered with various facets such as top-related tags in the result set, top categories, top users for the query, dominant timeline, etc.
- On the faceted interface, the user can get a glimpse of the underlying data attributes and can filter the result with each iteration.
- Clicking on a single thumbnail will lead to a video detail page (navigational search) where the video is displayed not only with the original descriptions, but also with some extra resources related to the user query concept.
- These resources are connected to the user query concept. There may be too many resources in one DBpedia page and all are not of equal relevance. In future, we need to figure out how to rank the connected resources in relation to the query concept. One heuristic may be to rank the resources of a similar type higher compared to the others, or we can compute a resource distance based on mutual information sharing such as categories, property values, etc. This part of the work is still ongoing.

V. Evaluation

Since the evaluation is still ongoing at the time of writing, we report a part of the evaluation. The objective here is to evaluate the effectiveness of the automatic augmentation of light semantics from various sources and its impact on retrieval in terms of user satisfaction.

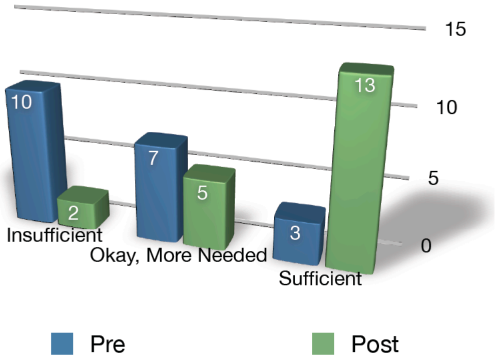


Figure 4. Evaluation of task 1 (content enrichment).

Effectiveness and user satisfaction are both measured qualitatively based on user ratings. Five users evaluated 20 random videos for their content description sufficiency. Each user was presented with a list of inferred keywords for describing the video content and were asked to rate the list for degree of sufficiency on a three-point scale of 1 to 3, after watching the video. The average video duration in the evaluation was 3.25 minutes.

A rating of 1 is the least descriptive (insufficient or irrelevant), while 3 is rated as a sufficient description of the depicted content, and a rating of 2 is considered as representing that there are some descriptions but more are needed. The result is based on inter-user agreement of ratings (a minimum of 3 out of 5 users agreed for a score).

Figure 4 shows the results of the evaluation of task 1, where the number of sufficient content descriptions increases to 13 videos from only 3 videos, whereas 5 videos are still considered to be in need of more descriptive keywords. The average rating per video increased from 1.65 to 2.5. In the evaluation of task 2, we have started to measure the level of user satisfaction for search results after enrichment.

VI. Conclusions and Future Work

We have discussed a lightweight framework to provide metadata for user videos on the Web using several existing ontologies. We discussed an approach to create instance data based on our models from user-generated content using both statistical and linguistic approaches.

We also described our approach to integrate the structured video data into the Linked Open Data cloud for greater integration and interoperability. Finally, the paper details an implemented prototype for the semantic search of web videos with three different modes of user interface.

Our future work involves robust evaluation of the instance-learning module and the creation of a fully-fledged integrated semantic annotation and search system.

Acknowledgment

This work was supported by Science Foundation Ireland under grant number SFI/08/CE/I1380 (Líon 2).

REFERENCES

[1] C. Mangold, “A survey and classification of semantic search approaches”, Int. J. Metadata, Semantics and Ontology, vol. 2, pp. 23-34, 2007.

[2] D. T. Tran, S. Bloehdorn, P. Cimiano, and P. Haase. “Expressive resource descriptions for ontology-based information retrieval”, Proc. of the 1st Int. Conf. on the Theory of Information Retrieval (ICTIR ‘07), October 2007.

[3] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello, “Sindice.com: a document-oriented lookup index for open linked data”, IJMSO, vol. 3, no. 1, pp. 37-52, 2008.

[4] R. Arndt, R. Troncy, S. Staab, L. Hardman, and M. Vacura, “COMM: Designing a Well-Founded Multimedia Ontology for the Web”, Proc. of the 6th International Semantic Web Conference (ISWC 2007), Busan, Korea, November 11-15, 2007.

[5] J. Hunter, “Adding multimedia to the Semantic Web – Building an MPEG-7 ontology”, 1st International Semantic Web Working Symposium (SWWS ’01). California, USA, pp. 261–281, 2001.

[6] P. Mika, “Ontologies are us: a unified model of social networks and semantics”, ISWC 2005, LNCS vol. 3729, pp. 522–536, Springer, 2005.

[7] MPEG-7: Multimedia Content Description Interface, ISO/IEC 15938, 2001.

[8] A. Passant, “Using ontologies to strengthen folksonomies and enrich information retrieval in weblogs”, International Conference on Weblogs and Social Media, 2007.

[9] C. Schmitz, A. Hotho, R. Jäschke, and G. Stumme, “Mining association rules in folksonomies”, Data Science and Classification, pp. 261–270, 2006.

[10] L. Specia, E. Motta, “Integrating folksonomies with the semantic web”, Proc. of the 4th European Conference on the Semantic Web: Research and Applications, Innsbruck, Austria, 2007.

[11] C. Bizer, T. Heath, K. Idehen, and T. Berners-Lee, “Linked data on the Web,” Proc. of the 17th Int. Conf. on World Wide Web, ACM, 2008, pp. 1265–1266.

[12] S. Choudhury, J. Breslin, and S. Decker, “A lightweight web video model with content and context descriptions for integration with linked data”, Proc. of Semantic Authoring, Annotation and Knowledge Markup Workshop, 2009.

[13] J.G. Breslin, A. Harth, U. Bojars, and S. Decker, “Towards semantically-interlinked online communities”, Proc. of the 2nd European Semantic Web Conference (ESWC ‘05), LNCS vol. 3532, pp. 500-514, Heraklion, Greece, 2005.

[14] D. Brickley, L. Miller, “The Friend Of A Friend (FOAF) vocabulary specification”, http://xmlns.com/foaf/0.1/, 2005.

[15] C. Cattuto, D. Benz, A. Hotho, and G. Stumme, “Semantic grounding of tag relatedness in social bookmarking systems”, Proc. of the 7th International Semantic Web Conference, 2008.

[16] H. Halpin, V. Robu, and H. Shepherd, “The complex dynamics of collaborative tagging”, WWW 2007, ACM Press, 2007.

[17] L. Ding, T. Finin, A. Joshi, R. Pan, R.S. Cost, Y. Peng, P. Reddivari, V.C. Doshi, and J. Sachs, “Swoogle: A search and metadata engine for the semantic web”, Proc. of the 13th ACM Conference on Information and Knowledge Management, 2004.

[18] J. Waitelonis, H. Sack, “Augmenting video search with linked open data”, Proc. of Int. Conf. on Semantic Systems, 2009.

[19] M. Schraefel, A. Wilson, A. Russell, and D. A. Smith, “mSpace: improving information access to multimedia domains with multimodal exploratory search”, Commun. ACM, vol. 49, no. 4, pp. 47–49, 2006.

[20] E. Prud’hommeaux, A. Seaborne, “SPARQL query language for RDF”, W3C, 2008.

[21] Sigma: Available online at http://sig.ma

[22] Wikipedia: Available online at http://www.wikipedia.org

[23] YouTube: Available online at http://www.youtube.com

[24] Vimeo: Available online at http://www.vimeo.com

[25] W3C Media Annotation Group: Available online at http://www.w3.org/2008/WebVideo/Annotations/

[26] DBpedia: Available online at http://dbpedia.org

[27] OpenCalais: Available online at http://www.opencalais.com

A Semantic-based Recommender System Using A Simulated Annealing Algorithm

Romain Picot-Clémente, Christophe Cruz, Christophe Nicolle
LE2I, UMR CNRS 5158
University of Bourgogne, Dijon, France
{romain.picot-clemente, christophe.cruz, christophe.nicolle}@u-bourgogne.fr

***Abstract*— A recommender system based on semantic web technologies and on an adaptive hypermedia architecture is shown in this paper. The system uses a stochastic algorithm to provide recommendations to users. The paper presents the system architecture based on the semantic Web technologies and explains a simulated annealing algorithm performing the recommendations. A mobile application for the tourism domain proving the feasibility of this system is described at the end of the paper, some benchmarks are presented. In this application, the recommendations are defined as combinations of tourism products, which are linked to each other. The paper is mainly focused on the architecture and the recommendation process of the system.**

Keywords - ***Semantic based recommender system, adaptive hypermedia system, simulated annealing algorithm, tourist travel.***

I. INTRODUCTION

These last years, the number of customer relationship management (CRM) implementations has increased enormously. CRM systems aim at allowing organizations to provide fast and efficient user-focused services. A CRM system uses client related information or knowledge to provide relevant products or services to clients [1]. The increasing use of digital technologies by customers, and particularly the Web and mobile devices, is changing what is possible and what is expected in terms of customer management. CRM evolved from business processes such as the need to improve the client retention by the effective management of customer relationships [2].

Our project aims to facilitate tourists for the definition of a complete journey on the region Côte d'Or in Burgundy, France from a database composed of more than 4 thousand geo-localized tourism products. Today, searching and finding relevant tourism products related to a user profile is tedious. Consequently, a recommendation system has been defined. The use of personalized recommender systems [3] [4] [5] to assist customers in the selection of products is becoming more and more popular and wide-spread. Most of the recommender systems is based on algorithms computing recommendations using methods like collaborative filtering [6] [7], content-based classifier [8] [9] and hybrids of these two techniques [10] [11] [12].

Recommender systems suggest information sources and products to users based on learning from examples of their likes and dislikes [4]. A typical recommender system has three steps: 1/ Users provide examples of their tastes. These can be explicit, like demanding ratings of specific items, or implicit, like analyzing his browsing behavior. 2/ A user profile is computed using the information from the first step. It is a representation of the user's likes and dislikes; 3/The system computes recommendations using these user profiles.

Content-based (CB) and collaborative filtering (CF) methods are two of the main approaches used to form recommendations. Hybrid techniques integrating these two different approaches have also been proposed. The CB method has been based on the textual filtering model described in [13]. Generally, in CB systems, the user profile is inferred automatically from documents' content that the user has seen and rated. The profiles and domain documents are then used as input of a classification algorithm. The documents which are similar (in content) to the user profile are considered interesting and are recommended to the user.

CF systems [6] [7] are an alternative to CB systems. The basic idea is to go beyond the experience of an individual user profile and instead to use the experiences of a population or community of users. These systems are designed with the assumption that a good way to find interesting content is to find people with similar tastes and to propose items they like. Typically, each user is associated to a set of nearest-neighbor users, comparing profiles' information. With this method, objects recommendations are based on similarities of users rather than the similarities of objects.

Both CF and CB systems have strengths and weaknesses. In CF systems, the main problem is that the new objects with no rate cannot be recommended. CB systems suffer from deficiencies in the way of selecting items for recommendation. Indeed, the objects are recommended if the user has seen and liked similar objects in the past. Consequently, a variety of hybrid systems have recently been developed: 1/ Some use other users' ratings as additional features in a CB system [10]. 2/ Some use CB methods for the creation of bots producing additional data for "pseudo-users". These data are combined with real users' data using CF methods [12]. 3/ Others use CB predictions to "fill out" the probable user-items' ratings in order to allow CF techniques to produce more accurate recommendations [11].

We have developed a CB system inspired by Adaptive Hypermedia systems. Adaptive hypermedia systems are hypermedia systems (websites, e-learning platforms, etc.) with adaptive behavior to provide adaptive content, presentation and navigation to users, based on their knowledge, preferences, goals, etc. The purpose of the proposed system is to find the best combination of individuals from a domain ontology that fit to the user interest and we propose the use a simulated annealing algorithm to do this. The first part explains what an adaptive hypermedia system is. It is also shown in this part how adaptive hypermedia systems are positioned relative to recommender systems. Then, a part describes the architecture, the properties and the recommendation process of the proposed recommender system applied to tourism domain. The next part shows a utilization example for a touristic journey proposition and the final one gives some benchmark of the application.

II. ADAPTIVE HYPERMEDIA SYSTEMS

The research domain of adaptive hypermedia has been very prolific these 10 last years. Some systems [14] [15] [16] have been developed, giving principally solutions for e-Learning which is considered as the first application domain. Each system brings its own architecture and methods. Moreover, few attempts have been made to define reference models [17] [18] [19] [20] but without success because of being not enough generic to take account of the new trends and innovations. Nevertheless, most of the systems and models are based on a set of layers, also called models, which separate clearly the different tasks. Then, we can see that there are at least three models in common, necessary and sufficient to achieve adaptive hypermedia systems according to Brusilovsky [21]. It needs to primarily be a hypermedia system based on a domain. The domain model is a representation of the knowledge on a given subject the creator wants to deliver. It describes how the domain is organized and interconnected. The second model is called a user model which is a representation of the user within the system. It models all user information which may require the system to provide an adaptation. The last model is the adaptation model. It performs all the adaptive algorithms based on other models to provide an adaptation to the user. Beyond the use of domain, user and adaptation models, the trend is to use additional models like presentation, goals, context or other models. This allows to better identify the different performed tasks and to facilitate the construction of adaptive hypermedia systems. Nevertheless, there is no generic model integrating them for the moment.

Methods to model domain/user (adaptation principles are also described):

The **keywords vectors space** methods consider that each document and user profile is described by a set of weighted keywords vectors [22] [23] [24]. At the adaptation model, the weights are used to calculate the similarity degree between two vectors and then to propose relevant document to the user. The keywords representation is popular because of its simplicity and its efficiency. Nevertheless, the main drawback is that a lot of information is lost during the representation phase.

In **semantic networks**, each node represents a concept. Minio and Tasso [25] present a semantic networks based approach where each node contains a particular word of a corpus and arcs are created following the co-occurrences of the words from connected nodes into the documents. Each domain document is represented like that. In simple systems using only one semantic network to model the user, each node contains only one keyword. The keywords are extracted from pages which the user gives its taste. Then, they are treated to keep only the most relevant ones and are weighted in order to remove those with a weight lesser than a predefined threshold. The selected keywords are then added to the semantic network where each node represents a keyword and each arc their co-occurrence into the documents. With this method, it is possible to evaluate the relevance of a document compared to the user profile. Indeed, it suffices to construct a semantic network of a document and compare it to the semantic network of the user to classify it to interesting, uninteresting or indifferent documents.

Ontology approach is similar to the semantic network approach in the sense that both are represented using nodes and relations between nodes. Nevertheless, in concepts based profiles, nodes represent abstract subjects and not word or set of words. Moreover, links are not only co-occurrence relations between words, they have several significations. The use of ontology can keep a maximum of information. In QuickStep [26], the ontology is used for the research domain and has been created by domain experts. The ontology concepts are represented as vectors of article examples. The users' papers from their publication list are modeled as characteristic' vectors and are linked to concepts using the nearest neighbor algorithm. These concepts are then used to form a user profile. Each concept is weighted by the number of papers linked to it. Recommendations are then made from the correlations between the current interests of the user to topics and papers that are related to these topics. In [27] and [28], a predefined ontology is used to model the domain. User profiles are represented with a set of weighted concepts where weight represents the user's interest for a concept. Its interests are determined by analyzing its behavior.

Three types of adaptation have been highlighted in the researches on adaptive hypermedia systems: content, navigation and presentation adaptation. The content adaptation consists in hiding/showing or highlighting some information. The adaptation model makes the decision of which content has to be adapted and how to display it. The navigation adaptation consists in modifying the link structure suggesting links or forcing the user to follow a destination. There is URLs' adaptation or destinations adaptation. In the first, the adaptation model provides destination links to the presentation model; these links are displayed at the page generation. Whereas, in the second one, the adaptation model provides links without fixed destination to the presentation model; the destination is decided by the adaptation model when the link is accessed by the user. The presentation adaptation consists in insisting (or not) on the content parts or on the links. It consists also in adapting the preferences setting to the device or the page. The adaptation model process makes the decision of which content or links to insist in following the presentation context. Even if recommender systems are often differentiated from adaptive hypermedia systems, a lot of similarity between these two types of systems can be highlighted. Indeed, the recommender systems provide recommendations using different algorithms, as it is done in the adaptation model. Moreover, we can see that they model also users' tastes and domain's items, as it is done in adaptive hypermedia systems with the user model and domain model. Nevertheless, recommender systems perform only adaptation of the content whereas adaptive hypermedia systems realize two more adaptation types. Following these observations, a recommendation system appears to be a constrained adaptive hypermedia system. Thus, it seems clear that recommender systems can be defined as a subset of adaptive hypermedia systems, whatever its type (CB or CF).

The use of an adaptive hypermedia architecture for the creation of recommender systems is interesting because we can clearly define the tasks associated with each part of the application, and it gives the opportunity to evolve the system adding modules and/or other types of adaptation without difficult modifications of parts already implemented. For instance, a CB recommender system could be improved with features of CF systems, adding a group model where clusters of users can be defined. For the creation of our CB recommender system, we base on adaptive hypermedia architecture. Beyond the use of the three main ones (domain, user and adaptation model), a goal model has been added. It allows the modeling of users' goals. A description of the architecture is explained in the following part.

III. THE PROPOSED RECOMMENDER SYSTEM

This part describes, first, the architecture of the proposed recommender system. Then, the recommendation process is explained, it is based on a simulated annealing algorithm. It is followed by an overview of the implementation for the tourism application.

A. Architecture

The proposed recommendation system is based on a set of layers (models). It consists of a user model, a domain model, an adaptation model and a goal model. This modeling allows a clear separation between the tasks. The domain model defines the whole domain knowledge. It consists of sets of domain concepts and relations between concepts. Generally, the concepts index the contents or the pages in order to be provided to the user. The most appropriate structure we have chosen for this modeling is the ontology. Actually, it facilitates the creation of complex structures. It allows also the inferences and this structure is portable thanks to the standardized OWL language. The ontology concepts are populated by individuals representing instances of these concepts which can be provided to the user.

The goal model is an overlay model on the domain model. Actually, it consists of a set of goal concepts that bring together individuals of the domain model. A goal concept is a set of domain model individuals, knowing that different goal concepts can group same individuals. A goal is defined by an SWRL rule allowing the selection of the individuals which verifies the rule.

The user model aims at modeling the user into the system. In the present case, it is composed of two parts. The first part is based on the goal model which is based, by definition, on the domain model. This part is called overlay part on the domain or domain dependant part, or even dynamic part because it is very changeable. Instead, the second part is called static part; it is a domain independent part. The user model is composed of a set of goals concepts, selected using the user behavior and <attribute-value> pairs for data such as date of birth, gender, etc. With the dynamic part, we have an idea of the user interests on domain individuals. Actually, when an individual appears into more than one goal concept selected by the user, then this individual is considered more important for the user. Thus, we can induce interest weights on the domain individuals related to the selected goal concepts. Moreover, we can propagate these weights to the entire domain model using the links into the domain ontology.

The adaptation model is considered as the core of the system, the adaptive algorithms are carry out in this level. The recommendations provided to the user are formulated as a combination of individuals from the domain model, according its user model. The problem consists in finding the optimal combination of individuals from the domain model constrained by a user model. Browsing all the possible combinations to find the best one is not possible in a short time, consequently, we propose the use of a stochastic algorithm called simulated annealing in order to find a combination which is close to the best one in a short time. Simulated annealing [29] is an optimization technique particularly well suited to overcoming the multiple minima problem. Unlike gradient-descent methods, simulated annealing can cross barriers between minima and thus can explore a greater volume of the parameter space to find better models in deeper minima.

This algorithm is used to minimize an energy function defining the relevance of a combination according to a user model. This energy depends mainly on the interest weights deduced from the user dynamic part on the domain individuals. But, depending on the type of application, more parameters can be taken into account. For instance, we can use geographic parameters for an application which aims to provide a nearby restaurant corresponding to the user requirements and coordinates. Moreover, constraints can be defined in the ontology between individuals or/and concepts. For instance, a medical application which provides a combination of medicines has to indicate in the ontology when one medicine cannot be given with another one, so that the application cannot generate a bad combination.

B. Recommendation process

The recommender system aims at providing a combination of individuals from the ontology. This part presents how this recommendation is undertaken.

In order to solve the problem of providing the best combination of individuals from the domain model, we propose the use of a stochastic algorithm called simulated annealing for its resolution. Actually, this kind of algorithms allows to find a solution that approaches the best (or is the best) in a very short time. The simulated annealing is inspired from a method used in the steel industry. To obtain a metal with a perfect structure crystal type (fundamental state corresponding to the minimum internal energy), the process is as follow: after bringing the material to liquid, the temperature is lowered to solidification state. If the decrease of temperature is very sudden, a "glass" is obtained, feature of the technique of "hardening". On the contrary, if it is very gradual, allowing time for atoms to reach statistical equilibrium, it will tend toward more regular structures, to finish in the ground state: the "crystal", characterizing the system freeze. If this lower of temperature is not slow enough, defects could appear. Then, it would be necessary to correct them by heating the material again slightly to allow atoms to regain freedom of movement and facilitating a possible rearrangement toward a more stable structure.

The simulated annealing algorithm used into the adaptation model is based on this principle. At the beginning, the algorithm chooses an initial random combination of individuals following a given pattern (for instance, a combination consisting of a hotel, two restaurants and two activities). This combination has an energy E_0, called the initial energy, which represents the quality of a combination. The lower the energy is, the better the combination is. A variable T, called temperature, decreases in increments over time. At each level of temperature is tested a number of elementary random changes on the current combination. A cost d_f is associated to each modification; it is defined as the difference between the combination's energy after the modification and the one before. A negative cost signifies the current combination has a lesser energy than the previous one (thus better by definition), it is then kept. Conversely, a positive cost represents a "bad" change. Nevertheless, it can be kept according a given probability (acceptance rate t_a) depending on the temperature and the cost. The higher the temperature is, the higher the probability is. Thus, over time, the number of changes allowed decreases as the temperature decreases, until no longer accepting any changes. Finally, the system is said frozen, and the current combination becomes the final combination to be presented to the user. The acceptance rate is defined in (1) where T_k is the temperature at the level k, $k \in$ N.

$$t_a = e^{-\frac{d_f}{T_k}} \qquad (1)$$

$$T_0 = \frac{d_{fmean}}{\ln \frac{1}{t_a}} \qquad (2)$$

An initial temperature T_0 is computed using this formula and setting the values of the acceptance rate and the cost. The initial acceptance rate is defined arbitrarily and the cost is set calculating the average cost by performing multiple changes on random combinations. Thus, the initial temperature is presented in (2) where d_{fmean} is the average cost of the modifications.

The temperature decrease is achieved through a geometric decay at each level:

$$T_k = g(T_{k-1}) = coef \times T_{k-1} = coef^k \times T_0 \qquad (3)$$

where k is the current level and $0<coef<1$.

The relevance of a combination is determined by an energy function. The quality of the final combination, given by the simulated annealing algorithm, depends a lot on the definition of this energy. This function is highly dependent on the type of application. For instance, in a tourism application, the individuals and user coordinates could be taken into account, whereas these data are useless in a medical application. Nevertheless, the energy function is based, in all cases, on the user interests deduced from its dynamic part.

The dynamic part is constituted of goals determined by the user clicks on icons which are linked to goals. Thus, each time a user clicks on an icon, the related goal is added to the dynamic part of its user model. To deduce the user's interest weights on the ontology individuals, an algorithm of weight propagation uses the fact that each goal is a set of rules including individuals from the domain model. Thus, each time an individual is selected by a rule, its weight is incremented. Therefore, this weighting allows the demarcation of some individuals, giving an idea of the user interests. With this modeling, after few user clicks on icons, the system can quickly provide a combination of domain individuals that matches its interests. According the definition, this type of recommender system is a CB system. Nevertheless, it is also possible to base on group of users to have the benefits of CF systems. It just needs to add a group model to the system. The next part shows an application of this modeling for a tourism application which is currently in development.

C. *Tourism implementation*

This modeling is being applied to the tourism domain in the region of Côte-d'Or in France for the company Côte-d'Or Tourisme. The aim is to create a tourism application that should provide a combination of tourism products from Côte-d'Or according to a user profile. At the beginning, a domain ontology has been created with all the concepts and the individuals related to the application domain. This ontology was supplied from a database composed of more than 3000 tourism products. Then, a goal model has been defined using goal concepts like "Week end", "Going out with friends", "with a baby", etc. This knowledge was generated from the specialists of the domain represented by people working for the company Côte-d'Or Tourisme.

An empirical pattern is defined to determine what kind of combination the adaptation model has to return. The energy function which gives the relevance of a combination is based on the interest weights and the coordinates of the tourism products, because it is not relevant to propose an activity in the morning and a restaurant for lunch with a distance of more than 50 kilometers. The traveling time required to reach the restaurant after the activity ending is inappropriate.

A variance threshold needs to be set in order to define the maximum preferred variance between the individuals coordinates. This variance characterized the value dispersion regarding the average, in this case the threshold. The subsets in this pattern are possible. For instance, we can define a pattern like “Accommodation, Restaurant1, Activity, Restaurant2” in which “Accomodation and Restaurant1” are the first subset, and “Activity and Restaurant2” the second subset. In addition, a variance threshold is defined for each one. Thereby, the system can use more complex patterns for the combinations.

The variance of a combination is defined as follow:

$$\operatorname{var}(C) = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}\left(\left(C_{ix} - \frac{1}{N}\sum_{i=0}^{N-1}C_{ix}\right)^2 + \left(C_{iy} - \frac{1}{N}\sum_{i=0}^{N-1}C_{iy}\right)^2\right)^2} \quad (4)$$

where C is a combination, N the number of elements into the combination, C_i the ith element of the combination, and C_{ix} and C_{iy} the x and y coordinates of the ith element.

The weight of a combination is defined as follow:

$$weight(C) = \frac{1}{N}\sum_{i=0}^{N-1} C_{iweight} \quad (5)$$

where $\mathrm{C}_{\mathrm{iweight}}$ is the weight of the ith element.

Using the variance and the weight function, the energy of a combination is:

$$Energy(C) = \frac{1}{weight(C)} \times E\left(\frac{\operatorname{var}(C)}{Threshold_C}\right) \times \prod_{j=0}^{L-1} E\left(\frac{\operatorname{var}(G_{Cj})}{Threshold_{G_{Cj}}}\right) \quad (6)$$

where $E(X)$ is the integer part of X, $Threshold_C$ is the variance threshold of the geographic coordinates for the combination C, $Threshold_{G_{Cj}}$ is the variance threshold of the geographic coordinates for the jth subset of C, and L the number avec subsets. Thus, the system performs the simulated annealing algorithm using this energy function and a user profile, so that it can provide a combination of tourism products matching its interests and close coordinates. The result of the algorithm gives a combination of close products with high weights. The next part shows an example of a touristic journey provided by this implementation. An interface has been developed for iPhone.

IV. AN UTILIZATION EXAMPLE

An interface for this tourism implementation has been developed for iPhone. This part explains briefly the utilization of this application. The user is first invited to define his profile by giving his stay duration and by clicking on goal icons in order to inform on its interests. Moreover, the geographic coordinates of the user can be used or specific geographic coordinates can be specified for a preferred area. Nevertheless, if no area is given, the area will be the entire region of Côte-d’Or. Tourism products can be also selected and a complete stay will be generated relevantly according these selections. After this step, the adaptation process is performed using the simulated annealing algorithm and the system provides a combination of tourism offers, corresponding to the user's profile, in a carousel. If the solution does not satisfy the user, he is able to demand a new generation, keeping some elements if wanted. Thus, the system takes the kept elements into account to provide a new combination. This new solution is generated by fixing the kept elements into the combination. Thus, only the others elements of the combination are modified during the process of researching the best combination. The kept elements are only considered for the energy computations. A benchmark is presented in the next part to show the relevance of the simulated annealing algorithm.

V. BENCHMARK

Some tests of the algorithm for the generation of combinations have been done on a set of three thousand tourism products. We did comparisons between the energy of random combinations, the energy of the solutions found by the algorithm and the energy of the optimal combination. The solutions given by the simulated annealing algorithm are closed or equal to the optimal solution in terms of energy. In these tests, the average time required for the generation of a combination of six products was around 3 seconds. But, this time depends on the different parameters (temperature decrease rate, the number of iteration per level of temperature) necessary to perform the simulated annealing algorithm. The faster is the temperature decrease and the lower is the number of iterations, the faster is the generation, but the worse is the resulting combination. In any case, this time is better than the time required to find the best combination by browsing all the possibilities. For example, in our test, finding the best combination needed around 3 hours against 3 seconds using the simulated annealing algorithm. These times are only given to have orders of magnitude, more tests need to be performed to have exacts results and to prove the interest of our proposition. Nevertheless, given these few results, the algorithm seems to give a relevant solution according a predefined energy function with a lesser cost (in terms of time) than calculating the optimal solution.

VI. CONCLUSION

In this article, we presented a new content based recommender system in order to improve the customer relationship management in e-tourism. The idea consists to take advantages of the semantic Web technologies, the properties of adaptive hypermedia systems, and to combine them with combinatory algorithms in order to create a recommender system. The simulated annealing algorithm is used in order to solve the problem of the polynomial time search required to generate a combination of tourism products. It gives a solution which approaches the best solution in a short time. The few results seem to be good considering the time required to obtain them and comparing to the best solutions. Nevertheless, for the future, we need to make more tests and benchmarks to quantify more precisely the relevance of our system. Moreover, we could improve the quality of the propositions by taking into account some group of users as it is done in collaborative filtering recommender systems. This is possible by adding a group model into the architecture. Thus, the recommender system would become a hybrid recommender system.

ACKNOWLEDGMENT

This research is done in collaboration with Côte-d’Or Tourisme, a company which aims to promote tourism in its region. The authors would like to thank Quentin Collin and Erwan Privat for their help.

REFERENCES

[1] Levine, S. 2000. The rise of CRM, *America’s Network*, Vol. 104, No. 6, pp. 34.

[2] Bull, C. 2003. Strategic issues in customer relationship management (CRM) implementation, *Business Process Management Journal* 9, No. 5, pp. 592-602.

[3] Tintarev, N. and Masthoff, J. 2007. A survey of explanations in recommender systems, *ICDE'07 Workshop on Recommender System*, pp. 801-810, Istanbul, Turkey.

[4] Bilgic, M. and Mooney, R. J. 2005. Explaining recommendations: Satisfaction vs. promotion. In *Beyond Personalization Workshop*, IUI’05, pp. 13-18.

[5] Mcsherry, D. 2005. Explanation in recommender systems. *Artificial Intelligence Review*, Vol. 24, No. 2, pp. 179-197.

[6] Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the Association for Computing Machinery*, Vol. 35, No. 12, pp. 61-70.

[7] Herlocker, J., Konstan, J., Borchers, A., and Riedl, J. 2000. An algorithmic framework for performing collaborative filtering. In *Proceedings of 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 230-237, Berkeley, CA.

[8] Mooney, R. J. and Roy, L. 2000. Content-based book recommending using learning for text categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pp. 195-204, San Antonio, TX.

[9] Pazzani, M. J., Muramatsu, J., and Billsus, D. 1996. Syskill & Webert: Identifying interesting web sites. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pp. 54-61, Portland, OR.

[10] Basu, C., Hirsh, H., and Cohen, W. W. 1998. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pp. 714-720, Madison, WI.

[11] Melville, P., Mooney, R. J., and Nagarajan, R. 2002. Contentboosted collaborative filtering for improved recommendations. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, pp. 187-192, Edmonton, Alberta.

[12] Sarwar, B. M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., and Riedl, J. 1998. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, pp. 345-354, Seattle, Washington, USA.

[13] Oard, D. 1997. The State of the Art in Text Filtering, *User Modeling and User Adapted Interaction*, Vol. 7, No. 3, pp. 141-178.

[14] Brusilovsky, P., Eklund, J., and Schwarz, E. 1998. Web-based education for all: A tool for developing adaptive courseware, *Computer Networks*, Vol. 30, No. 1-7, pp. 291-300.

[15] Cristea, A. and De Mooij, A. 2003. LAOS: Layered WWW AHS authoring model and their corresponding algebraic operators, In *Proceedings of WWW'03,* Budapest, Hungary.

[16] Henze, N. 2000. Adaptive hyperbooks: Adaptation for project-based learning resources. *PhD Dissertation*, University of Hannover.

[17] Cristea, A. and Calvi, L. 2003. The Three Layers of Adaptation Granularity. In *Proc. of the International Conference on User Modelling*, pp. 4-14, Johnstown, PA, USA.

[18] Hendrix, M. and Cristea, 2008. A. Meta-levels of adaptation in education, *Proceedings of 11th IASTED International Conference on Computers and Advanced Technology in Education*, V. Uskov (Ed.), IASTED, Innsbruck, Austria.

[19] De Bra, P., Houben, G-J., and Kornatzky, Y. 1992. An extensible data model for hyperdocuments, *Proceedings of the ACM Conference on Hypertext*, pp. 222-231, New York, NY: ACM.

[20] De Bra, P., Houben, G-J., and Wu, H. 1999. AHAM: A dexter-based reference model for adaptive hypermedia, in *Hypertext’99: Proceedings of the 10th ACMConference on Hypertext and Hypermedia: Returning to our Diverse Roots*, pp.147-156, New York.

[21] Brusilovsky, P. 2001. Adaptive hypermedia, *User modeling and user-adapted interaction*, Vol. 11 No. 1-2, pp. 87-110.

[22] Moukas, A. 1997. Amalthaea: Information Discovery and Filtering using a Multiagent Evolving Ecosystem. *J. Appl. Intell,* Vol. 11, No. 5, pp. 437-457.

[23] Kamba, T., Sakagami, H., and Koseki, Y. 1997. Antagonomy: A Personalized Newspaper on the World Wide Web, *Int’l J. Human-Computer Studies*, Vol. 46, No. 6, pp. 789-803.

[24] Armstrong, R., Freitag, D., Joachims, T., and Mitchell, T. 1995. WebWatcher: A Learning Apprentice for the World Wide Web, in *Proceedings of the 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford.

[25] Minio, M. and Tasso, C. 1996. User Modeling for Information Filtering on INTERNET Services: Exploiting an Extended Version of the UMT Shell. In *UM96 Workshop on User Modeling for Information Filtering on the WWW*; Kailua-Kona, Hawaii.

[26] Middleton, S., Alani, H., Shadbolt, N., and De Roure, D. 2002. Exploiting synergy between ontologies and recommender systems. In *Proceedings of the WWW international workshop on the semantic web*, Maui, HW, USA.

[27] Cantador, I. and Castells, P. 2006. A multilayered ontology-based user profiles and semantic social networks for recommender systems. In *Proceedings of the 2nd International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces (WPRSIUI 2006), at the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2006)*, Dublin, Ireland.

[28] Sieg, A., Mobasher, B., and Burke, R. 2007. Learning Ontology-Based User Profiles: A Semantic Approach to Personalized Web Search. In *IEEE Intelligent Informatics Bulletin*, Vol. 8, No. 1, pp. 7-18.

[29] Laarhoven, P.J.M. and Aarts, E.H.L. 1987. Simulated Annealing: Theory and Applications. D Reidel Publishing Company, Germany.

Hierarchical Organization of the Semantic Rules for the Images Annotation By Co-Quotation Method

Yassine Ayadi
MIR@CL Laboratory
Multimedia InfoRmation systems and @dvanced Computing Laboratory
University of Sfax, Tunisia
ayadi.yassine@gmail.com

Ikram Amous, Faiez Gargouri
MIR@CL Laboratory
Multimedia InfoRmation systems and @dvanced Computing Laboratory
University of Sfax, Tunisia
ikram.amous@isecs.rnu.tn, faiez.gargouri@gmail.com

***Abstract*— The present paper introduces an approach for image semantic annotation. It discusses work in progress and reports the current state of our approach. This comprises the development of the domain ontology used for annotation, the functionalities for annotating image with an underlying ontology and search features based on these annotations. We describe a method for automatic annotation of images and apply it to and evaluate it on images of inference process.**

Keywords-Automatic Annotation; Semantic; Co-Quotation; Ontology; Inference;

I. Introduction

During the last decades, a number of digital images have burst with the advent of digital cameras which require effective search methods. However, due to the semantic gap between image visual features and human concepts, most users prefer textual queries. Hence, it is always difficult to find a specific image if we want to show it or share it with another person. In this context, the use of annotation can facilitate the task of images management. Besides, the image annotation establishes the main tool for semantics associated with an image. Moreover, the addition of meta-data to an image enriches its description and allows the construction of more successful consultation tools and visualizations.

Our work objective is to describe the multimedia document contents, facilitate and optimize their search. To do so, we build on the documents annotation by semantics descriptors. With semantics, we imply any information that can be deduced and explicitly specified. We can deduce the Car in the parking without such information being directly mentioned in the document. According to [1], semantics depends on the knowledge level and on the user perception as well as on its objective. Therefore, the semantics of a situation (or of a context) can be differently expressed by diverse users.

Furthermore, labeling the semantic content of images with a set of keywords is a problem known as image annotation. Annotation is used primarily for image database management, especially those using keyword-based search, while not annotated images can be extremely difficult to find in large database.

Once the documents are annotated, they can be used such as [17]. Indeed, there exists much work on the multimedia documents manually annotation, among which we quote: AnnoSearch [2], IMAGINATION [3], IAM@Image CLEFPhoto Annotation [4].

In our study, the idea is to exploit the visual descriptors and topological relationships in image to determine their semantics. Actually, neither tool presents concepts of exactly annotated images.

The continuation of this paper is organized in the following way. First of all, Section 2 presents our proposed annotation approach. Then, Section 3 provides the use of urban ontology which will be used in this work. Afterward, Section 4 illustrates the construction of hierarchy semantic rules where we use the Co-quotation method in the images annotation. Next, Section 5 describes the automatic image annotation. As for Section 6, it presents an enrichment of ontology and process inference. Finally, this paper ends with some concluding remarks and future perspectives.

II. Proposed Approach

The automatic image annotation is an effective research subject [5]. Its goal is to develop methods that can produce, for a new image, the relevant keywords among an annotation vocabulary. These predictions of keywords can be used to propose the image semantics.

We will describe in this section our annotation approach based on three steps to solve these problems. The proposed approach is illustrated by Figure 1.

- The first step ***Training***: extract the image preliminary characteristics to classify objects (many tools exist permit to automatically associate with some image a characteristics vector).In our study, the idea is to exploit the visual descriptors and topological relationships in image to determine their semantics. Actually, neither tool presenting concepts annotates exactly the images. The existing tools do not combine the object detection and the relation one.

The annotation process, defined in the Training step, is composed of two sub-steps:

I. We start with a set of images, which we call the images of apprenticeship. We proceed in a way that the users select an object of the image manually. The selection of an object makes possible for the user to affect a manual semantics annotation. The tools for image processing determine the low level characteristics defining this object.

ISBN: 978-1-61208-104-5

II. The second sub-steps consist in building a value matrix of low level descriptors describing the required object. This matrix represents the result of several iterations of the first phase on the basis of image for the same object.

After the apprenticeship phase, the system automatically creates one or several rules describing the object chosen from the first phase.

The main objective of our classification is to associate a unique interpretation from low level descriptors with an image document [6]. The result of the combination of the MPEG-7 descriptors with those of cavities and contour is a well-formed XML file.

The spatial relationships constitute the basis of the linguistic descriptions of the spatial configurations. These relations are generally classified a different category [8], [9], [10]: topology, orientation and distance. These can be descended of an explicit declaration on the part of the user or inferred from the existing information. In our proposition, we build on topological relationships. To detect the relationship between two detected objects, we calculate the angle between the including rectangles.

In our context, several object types can be distinguished: car, building, persons, panels, road, etc. These different objects are classified according to classes: means of transport, buildings, the place and objects. In order to do that, we can determine a set of spatial relations that can exist between objects in a picture to know: right, left, behind, in front. [7]

- In the second step, the process describes the Image ***semantic annotation***. This step is the relationships extraction between objects, to build the first level semantic rules (these rules represent a human knowledge). They are stored in a knowledge base. From an archive picture, we loosened a semantic set, represented through the notion of predicate logic. From all the rules, we loosen the various predicates whose semantics is extracted from the image. These predicates can be grouped in elements establishing the rule in relation between elements and result.
- The third step consists in the ***Inference process*** creation to generate a new semantics. The inference is then defined, then, as an action which allows a human or a machine to increase its knowledge. This person or this machine “makes an inference”, i.e, it infers a result starting from a set of data. Our process of inference consists in a unit of inference rules, being based on the principle of the front chaining.

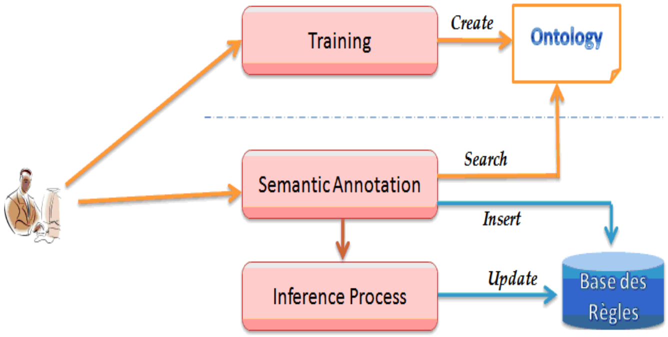


Figure 1. Proposed Demarche

To create a model, we ask to the domain expert to draft a list, as complete as that possible, in natural language, various semantics extracted of the images. He integrates into this list the knowledge environment and the studied context.

III. Urban Ontologies

Ontologies are used to formalize the concepts semantics of each domain. We have already used ontology of field formalized in XML (TOWNTOLOGY) which represents the concepts used by the urban image.

Ontology defines a common vocabulary for the researchers who need to share the information in a domain. It includes by the definitions legible by a machine on the basic concepts of this domain and of their relations. [11] [16]

Ontology is a formal explicit description of the concepts in a domain (classes), properties of every concept describing characteristics and attributes (facets).

Ontology as well as all the individual instances of the classes constitutes a knowledge base. The classes constitute the main concepts of several ontologies. They describe the concepts in the domain. A class can have subclasses which represent more specific concepts than the super-class (or superior class). The attributes describe the classes and the instances properties.

Let us illustrate these ideas for our domain. We are interested in the objects component of an image by describing the taxonomy of types represented by the model in Figure 1.

On this model, the Object class is subdivided into subclasses such as "Means", "Place", "Panel", "Buildings" and these in specialized subclasses as "Building", "car" or "Road"

This ontology also contains topological relations such as "Front", "Behind", "to the right", "to the left".

Properties are represented on the model by elementary principles, called Properties. If we assign precise objects in X and Y, these principles will become assertions: ' a car in front of a building '.

IV. Hierarchy of Semantic Rules

The manual annotation of semantics remains a problem, because it depends on the user objective and on his knowledge. It thus seems to us thus convenient to find a way of automating the annotation of semantics to improve the research for the multimedia documents by building on requests.

Several questions, thus, are left to be elucidated:

- How can we extract the semantic contents of the multimedia documents in an automatic way to ensure and facilitate future research for users?
- How can we connect high level knowledge to the low level characteristics of the documents?

The process of images annotation, which we propose, is based primarily on the results provided by Training step. The descriptors, extracted and instantiated automatically, from such a step, allows the construction of our first work scheme.

The result of this step consists in extracting automatically the low level descriptors and deducing the image elements recorded in ontology. Each element can have well determined semantics and can refer to an object (person, car, building…).

The combination of these elements with space relations [14] creates the first level of semantics rules describing the image content. This first level is created manually.

An image can refer to several semantic rules. By using this context, makes it possible to us to gather the semantic rules by topic.

The purpose of the Co-quotation method [12], used in bibliometry since 1973, is to create starting from scientific articles of the same field of research and by using their bibliographical references, of the relations between these articles.

This method rests on the assumption that two bibliographical references of unspecified dates, frequently quoted together, have a parity set of themes. In the same way that for the table of the couplings, the matrix of Co-quotations is built by each line is the studied set quotations i, each column is the quotations set j and the elements set x_{ij} of the matrix corresponds to the number of documents which quoted documents i and j in same time.

The use of the bonds in order to annotate a resource also applies to the images.

The use of the Co-quotation method in the images annotation can help us to use the annotations of close references by topic in order to annotate new images.

The following figure (Figure 2) is an extract of the quotation graph: the image ***I*** cite semantic rules ***RG***

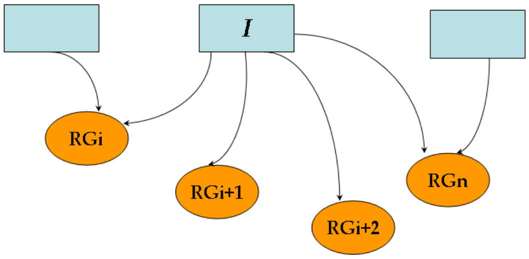


Figure 2. Extract of the Quotation Graph

The method of Co-quotation [12] is used to calculate the resemblance between the semantic rules and not between the images.

The following figure (Figure 3) is an example of graph of Co-quotation. Value 2 between rules $\boldsymbol{RG_i}$ and $\boldsymbol{RG_{i+2}}$ indicates that these two rules are quoted together by two images.

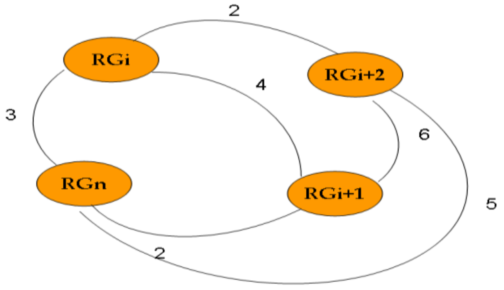


Figure 3. Example of Graph of Co-Quotation

The Co-quotation matrix is a representation of the Co-quotation graph; it corresponds to a square matrix.

C_{i+2n}: (i+2 Line, n Column) the Co-quotation frequency of $\boldsymbol{RG_{i+2}}$ and $\boldsymbol{RG_n}$, which is equal to 5, because they are quoted together by five Images.

$$\begin{array}{c|cccc} & \mathrm{RGi} & \mathrm{RGi+1} & \mathrm{RGi+2} & \mathrm{RGn} \\ \hline \mathrm{RGi} & 0 & 4 & 2 & 3 \\ \mathrm{RGi+1} & 4 & 0 & 6 & 2 \\ \mathrm{RGi+2} & 2 & 6 & 0 & 5 \\ \mathrm{RGn} & 3 & 2 & 5 & 0 \end{array}$$

Figure 4. The Co-quotation matrix

The result of the distance function [13]

$$Si,j = 1/C(i,j)2 \qquad (1)$$

Being in the interval [0,1]. Two semantic rules are quoted units, the more the distance S (I, J) will be close to zero.

As soon as the semantic rules of a document image are quoted, we build the graph of distance and the matrix of distance MC.

Matrix MC of the example is the following one:

$$\begin{array}{c|cccc} & \mathrm{RGi} & \mathrm{RGi+1} & \mathrm{RGi+2} & \mathrm{RGn} \\ \hline \mathrm{RGi} & 0 & 1/C(i,j)2 & 1/C(i,j)2 & 1/C(i,j)2 \\ \mathrm{RGi+1} & 1/C(i,j)2 & 0 & 1/C(i,j)2 & 1/C(i,j)2 \\ \mathrm{RGi+2} & 1/C(i,j)2 & 1/C(i,j)2 & 0 & 1/C(i,j)2 \\ \mathrm{RGn} & 1/C(i,j)2 & 1/C(i,j)2 & 1/C(i,j)2 & 0 \end{array}$$

Figure 5. The Co-quotation Distance

In order to represent association (Rules/Images), we use the concepts lattice. The concepts lattices are used in the search for information to refine or generalize the request user. [15]

The Galois lattice or concepts lattice is a mathematical structure making it possible to represent the not disjoined classes subjacent with an objects set [18].

Context: a context is a triplet K = (O, A, ζ) where O is an objects or individuals set, A is an attributes or properties set and ζ is a binary relation between O and A.

A context K = (O, A, ζ) can be represented in the table form, where a line corresponds to an object with its attributes.

Lattice: a lattice is an ordered Set in which two unspecified elements have an upper limit and a lower limit. A complete lattice is a lattice for which any element has an upper limit and a lower limit.

Galois correspondence: Is the context K = (O, A, ζ), f an application P (O) in P (A) and g an application P (A) in P (O), f and g defined:

- f: P(O)→ P(A) f (Oi) = $\{a \in A|\ (o,a) \in A, \forall o \in Oi\}$ intention ;
- g: P(A)→P(O) g(Ai) = $\{o \in O|\ (o,a) \in A, \forall a \in Ai\}$ extension ;

The couple (f, g) is called the Galois correspondence on K.

Formal concept: Are Oi $\subseteq$ O and Ai $\subseteq$ A, (Oi, Ai) is a concept if:

- Oi is the extension of Ai;
- Ai is the intention of Oi;

Oi = g (Ai) and Ai = f (Oi)

Galois lattices: Are f: O → A and g : A → O two functions defined on the lattices (O,$\leq$O) and (A, ,$\leq$A), such as (f, g) is a Galois correspondence.

Either G= {(o, a), where o is an element of O and where a is an element of A, such as o = g (a) and a = f (o)}. That is to say $\leq$ the relation of order defined by: (o1, a1) $\leq$ (o2,a2) if a1 $\leq$A a2. (G,$\leq$) is a Galois lattice.

Example:

The following table represents the correspondence between six images answers of the five rules {RG1, RG2, RG3, RG4, RG5}

The Rules are:

- RG1, Car ➔ Transport Means
- RG2, Taxi ➔ Car
- RG3, Car in front of Car ➔ Parking
- RG4, Taxi in front of Taxi ➔ Parking
- RG5, Transport Means in front of Transport Means ➔ Parking

TABLE I. IMAGE/ RULES ASSOCIATION

	RG1	RG2	RG3	RG4	RG5
I1	X	X			X
I2			X	X	X
I3	X			X	X
I4	X				X
I5				X	X
I6			X		X

The example of Galois correspondence is:

- O1 = {I3, I4} ➔ f(O1) = {RG1, RG5}
- A1 = {RG1, RG5} ➔ g(A1) = {I1, I3, I4}

In this example we have the couple ({RG5}, {I1, I3, I4})

In this example, we have the couple ({RG5}, {I1, I3, I4}) which means that the result of the request with rule RG5 gives for answer the images I1, I3, I4.

We illustrate the result of the Bordat algorithm [14] on the example of the preceding table.

C = (Ø, {I1, I2, I3, I4, I5})

δC = max {fC(I1), fC(I2), fC(I3), fC(I4), fC(I5), fC(I6)}
= max {{RG1, RG2, RG5}, {RG3, RG4, RG5}, {RG1, RG4, RG5}, {RG1, RG5}, {RG4, RG5}, {RG3, RG5}}
= max {{RG1, RG2, RG5}, {RG3, RG4, RG5}, {RG1, RG4, RG5}}

In this case the direct successors of C are:

C1 = ({I1}, {RG1, RG2, RG5})
C2 = ({I2}, {RG3, RG4, RG5})
C3 = ({I3}, {RG1, RG4, RG5}}

In the same way, one calculates the direct successors of C1:

δC1 = max {fC1(I1), fC1(I2), fC1(I3), fC1(I4), fC1(I5), fC1(I6)}
= max {{RG5}, {RG1, RG5}, {RG1, RG5}, {RG5}, {RG5}}

The C1 Successors are:

C4 = ({I1, I3, I4}, {RG1, RG5})

The continuation of the direct successor's calculation is made same manner. The result of the example is the following (Figure 6)

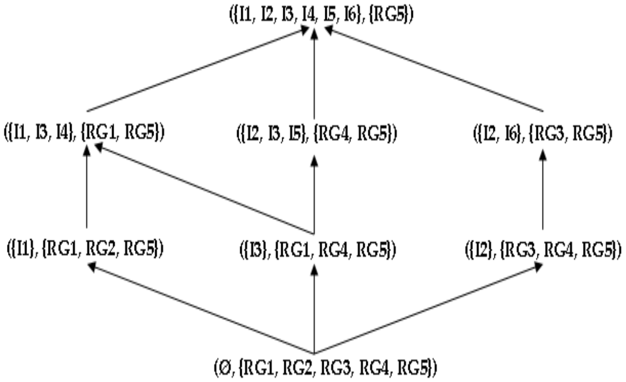

({I1, I2, I3, I4, I5, I6}, {RG5})
({I1, I3, I4}, {RG1, RG5})
({I2, I3, I5}, {RG4, RG5})
({I2, I6}, {RG3, RG5})
({I1}, {RG1, RG2, RG5})
({I3}, {RG1, RG4, RG5})
({I2}, {RG3, RG4, RG5})
(Ø, {RG1, RG2, RG3, RG4, RG5})

Figure 6. Galois Lattice

The lattice structure is used in order to extract the hierarchical relations between the semantic rules. We build the hierarchy of the semantic rules in order to keep only one occurrence of the rules. We leave the rules set of the more high level and one eliminates the occurrences from each element in the lower levels.

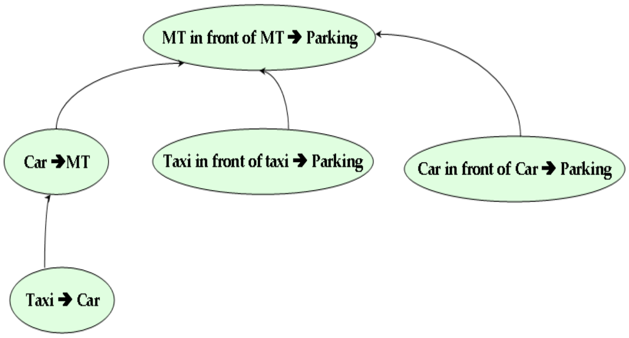

MT in front of MT → Parking
Car → MT
Taxi in front of taxi → Parking
Car in front of Car → Parking
Taxi → Car

Figure 7. Semantics Rules Lattice

Figure 7, present the result of the semantic rules hierarchical.

V. AUTOMATIC IMAGE ANNOTATION

The presentation of the imported annotations is made by defining a multi-criterion choice to select annotations to be used in the following phase.

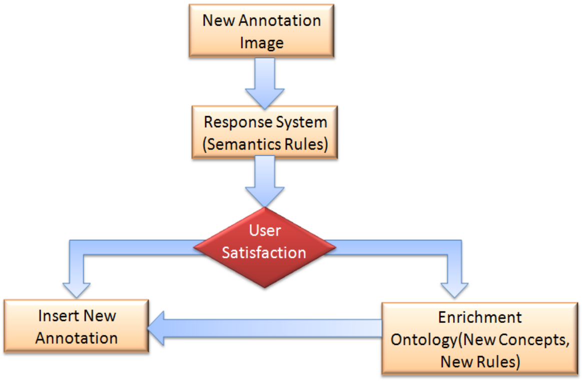

New Annotation Image
Response System (Semantics Rules)
User Satisfaction
Insert New Annotation
Enrichment Ontology(New Concepts, New Rules)

Figure 8. Automatic Image Annotation

Each Rule semantic has a rate (utilization Ratio of the rule for forthcoming annotations). This rate is a value included/understood an interval [0, 1].

The rate is calculated by the following formula:

$$\text{Fact-salt / Fact-Aff} \qquad (2)$$

Such as:

- ***Fact-salt***: It is the number of times that the rule at summer chosen as a solution for the annotation of a new image
- ***Fact-Aff***: It is the number of times that the rule at summer suggested as a solution for the annotation of a new image.

```
Algorithm: Automatic Annotation
Data
    C ← {$C_1$, $C_2$,…,$C_n$} Concepts Set of New Image
    (With Ontology)
    ReG ← {$RG_1$, $RG_2$, …, $RG_n$} Rules Set
    List : List of Rules Semantic
    U Type Rules Semantic
    A Type Rule
Begin
//Extract all the Rules for each concepts couples
List ← Extract_Rules ($C_i$, $C_j$);
U = List Beginning
Repeat
// extract all the low level rules than the rule U
A ← Extract_rules_lowlevel (U);
//Add the new rules at the list
End Automatic Annotation
```

The goal in this section is to import and order the semantic rules quoted by a document image I.

We retain the following criteria to order the annotations as a whole ***ReG***; The rate of selection of the imported rules. If the rule appears in several images, the largest rate of selection will be considered (maximum).

VI. ONTOLOGY ENRICHMENT AND INFERENCE PROCESS

Ontological engineering consists of the search for general, reusable, shareable and durable concepts to build a model of knowledge able to help people solve problems [MIZ 04].

Because our step of annotation is based on ontology, we also dealt with the problem of the enrichment of ontology. This enrichment will also be used to refine or enrich the automatic annotation by the documents multi-media as Image type.

We should note that ontology is a set of concepts connected by the relation of Specialization/ Generalization and other like synonymy.

The principle of enrichment in our step does not include/understand the suppression and the transformation of concepts, but earlier to add principle again is that of the semantic rules (semantic rules is a set of concepts connected to each other by relations which provide one or more semantics which can be the same concepts of ontology).

We were interested within the framework of our work in used ontology TOWNTOLOGY.

This ontology is described by the concepts of the urban field and the relations which connect them. Figure 9 illustrates an extract of this ontology:

- A node represents a concept, represented by a circle in the figure (for example the C1 concept).
- Concepts are connected by directed arcs defining the relation of Specialization/ Generalization, here the C2 concept is a specialization of the C1 concept.

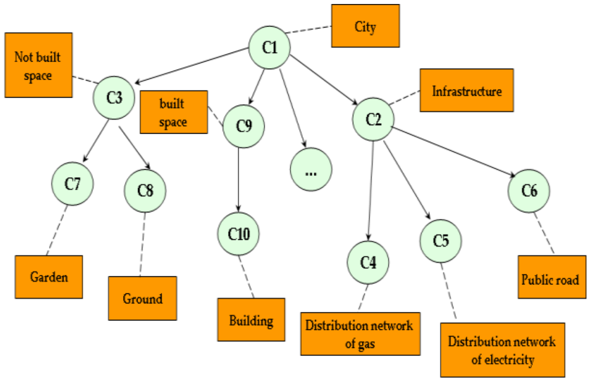


Figure 9. Example TOWNTOLOGY Ontology

The idea of our approach is based on the integration of the semantic rules within ontology. This stage of enrichment breaks up into two phases, the first stage is the manual integration of the rules within ontology; the second phase is an enrichment starting from the phase of interrogation.

- ***Manual Enrichment:*** Consist in adding semantic rules within descriptive XML file of ontology in order to build a first level of knowledge.
- ***Automatic Enrichment:*** is based on the exploitation of new annotations to enrich ontology.

Let us illustrate automatic enrichment for an example. A user ***U*** annotates a new image according to the following stages:

1. Initially, U seized the new image I, it obtains as an answer RG1, RG2… RG5 semantic rules.
2. The user can not be satisfied and it adds new concepts or semantic new rules
3. The U user finishes his annotation when it satisfies the result.

The increase in the information level which one can lay out on a system is essential for the improvement of this system control and the processes which are integrated there (automation, maintenance…).

Two primary sources give access to this information. The first one starts from human expert knowledge which gives rather qualitative information on what the studied system is, while the second is the data acquisition directly on the system, giving rather quantitative information.

The inference is an action which allows a human or a machine to increase its knowledge. This person or this machine “makes an inference”, i.e, it infers a result starting from a set of data.

Our inference process is composed of a set of inference rules, being based on the principle of the front chaining. In what follows a formal specification of all the inference rules as well as an application of these rules on the collected image basis are presented.

A. *The inference algorithm:*

```
Start
OPEN ← Semantic Rule
Repeat
     Ü ← Beginning OPEN
     Repeat
        To observe inference rules
        To add in end of OPEN semantic new Rules
     Until I = End Rules
     To add U to Close
Until OPNE = Ø
End
```

B. *Inference Rules*

Rule 1:

```
  BE: Elements Base
  BR: Relations Base
  BGR: Rules Base
  ∀ {OB1, OB2, S1, S2} ∈ BE
  ∀ R ∈ BR
  ∃ RG1 a Rule, RG2 a Rule /
       RG1 = OB1, R, OB2 → S1
       RG2 = OB1 → S2
       ∃ RG3 a new Rule /
       RG3 = S2, R, OB2 → S1
```

Rule 2:

```
  BE: Elements Base
  BR: Relations Base
  BGR: Rules Base
  ∀ {OB1, OB2, S1, S2} ∈ BE
  ∃ RG1 a Rule, RG2 a Rule /
       RG1 = OB1 → S1
       RG2 = S1 → S2
       ∃ RG3 a new Rule /
       RG3 = OB1 → S2
```

VII. Conclusion & Perspective

A system of research for image adapted to the needs of the users is capable of extracting the image semantics. However, the ditch between the low levels attributes and the semantic knowledge is the main obstacle in the construction of reliable semantics for the image research.

In this paper we proposed an approach which allows the discovery of semantic information from the low level image. Our approach is interested in the semantic description of the objects of a given image. We presented our vision for semantic annotation and inference to support the discovery of general image.

Our system is work in progress, and we are actively experimenting with implementation alternatives. As continuation of this work, the semantic representing the semantic and role relationships between the concepts will be constructed from our current sentence level semantic trees.

In this paper, we have described an interface for image annotation based on user-formulated semantic inference rules. The aim of this study was to determine the characteristics that suit the semantic inferencing and Rules. One of the significant findings was that knowledge of multimedia and image analysis terms is both a prerequisite and impediment to obtaining good results. We still found, however, that the results of applying rules defined by domain experts were significantly less than those defined by the authors.

References

[1] A. Boucher and T. Le. “Comment extraire la sémantique d’une image ?”. In 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, SETIT 2005, Tunisia. pp. 295-306. March 2005.

[2] X. Wang, L. Zhang, F. Jing, and W. Ma. “AnnoSearch : Image Auto-Annotation by search”. In. Computer Vision and Pattern Recognition, 2006. IEEE Computer Society Conference on (2006), pp. 1483 – 1490. June 2006.

[3] A. Walter and Gabor Nagypal. “The Combinaison of Techniques for Automatic Semntic Image Annotation Generation in the IMAGINATION Application”. In. ESWC 2008/ 5th European Semantic Web Conference. Tenerife, Sapin. 01- 05 June 2008. pp. 879-883. June 2008.

[4] J. Hare and P. Lewis. “IMA@Image CLEFPhoto Annotation 2009 : Naïve application of a linear-algebric semantic space Corfu”. In: CLEF 2009 Workshop, 30 September - 2 October 2009, Corfu, Greece. p. 66-71. October 2009.

[5] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. “Apprentissage de distance pour l'annotation d'images par plus proches voisins”. in: Reconnaissance des Formes et Intelligence Artificielle. CAEN. Inria-00439309, version 1. January 2010.

[6] Y. Ayadi, I. Amous, A. Jedidi, and F. Gargouri. "Towards a Semi-Automatic Description of the Multimedia Image Documents". In. The International Business Information Management Conference (8th IBIMA) on June 20, 21, and 22, 2007 in Dublin, Ireland. pp. 203-209. June 2007.

[7] Y. Ayadi, I. Amous, A. Jedidi, and F. Gargouri. "Automatic Annotation Process with Objects and Relationships Detection of Images". 5th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, SETIT 2009, Tunisia. pp. 56-61. March 2009.

[8] O. Bedel, S. Ferré, and O. Ridoux. "Handling Spatial Relations in Logical Concept Analysis To Explore Geographical Data". Int. Conf. Formal Concept Analysis (2008) pp. 241—257.

[9] A. Dia Miron, J. Gensel, M. Villanova-Oliver, and H. MARTIN. "Towards the Geo-spatial Querying of the Semantic Web with ONTOAST". 7th International Symposium on Web and Wireless GIS (W2GIS 2007), Cardiff, UK, 28-29 November 2007. pp. 121-136. November 2007.

[10] T. Leonard. "How language structures space". in H. Pick & L. Acredolo, éd., Spatial orientation: theory, research and application, New York, Plenum Press. 1983.

[11] A. Keita, R. Laurini, and C. Roussey. "Towards an Ontology for Urban Planning: The Towntology Project". In Proceedings of the 24th UDMS Symposium, Chioggia, October 27-29, 2004, pp. 12.-24.

[12] E. Garfield. "Co-Citation analysis of the scientific literature: Henry small on mapping the collective mind of science". Essays of an information scientist: Of Nobel Class, Women in Science, Citation Classics and Other Essays, 15 (19), 1993. pp. 293-303.

[13] L. Abrouk. "Annotation de documents par le contexte de citation basée sur une ontologie". Thèse de Doctorat. Soutenue le 27 novembre 2006. Université de Montpelier II.

[14] E. Mephu Nguifo and P. Njiwoua. 'Treillis de concepts et classification supervisée". Technique et Science Informatiques, 24(4) 2005 pp. 449–488

[15] N. Messai. "Treillis de galois et ontologies de domaine pour la classification et la recherche de sources de données génomiques". Master. University Henri Poincare. Juin 2004.

[16] A. Keita, C. Roussey, and R. Laurini. "Un outil d'aide à la construction d'ontologies pré-consensuelles : le projet Towntology". In actes du 24ème congrès de Informatique des Organisations et Systèmes d'Information et de Décision (INFORSID), Tunis 31 Mai-4 Juin 2006, pp. 911-926.

[17] A. Jedidi, I. Amous, and F. Sèdes. "Documents semi-structurés et métadonnées contribution à la réingénierie de collections de documents". Dans : RSTI ISI bases de données semi-structurées, Hermes - Lavoisier, Vol. 8, N. 5-6,. Decembre 2003.pp. 153-172.

Generating an Ontology Specific Editor

Hannes Niederhausen, Sven Windisch, Lutz Maicher
Topic Maps Lab
University of Leipzig
Leipzig, Germany
{*niederhausen, windisch, maicher*}@*informatik.uni-leipzig.de*

***Abstract*—Semantic technologies like Topic Maps provide a generic way of structured data representation. These technologies can be used to create data stores of any kind. To use them, it is necessary to fill them with data and therefore an editor is needed which provides input masks for the data. In this paper we present an Editor Generator Toolkit that enables developers to easily create small and fast editor applications for multiple platforms that allow easy collation of data.**

***Keywords*-Topic Maps, ontology, editor, TMCL, Eclipse RCP**

I. Introduction

Semantic technologies provide a generic way of structured data representation and thus are highly applicable as data stores for applications of all kinds. Beyond that, data stores must be filled with data and as most data is not available in an easily transformable format, an editor application is needed for the manual handling of the data.

In this paper, we present an *Editor Generator Toolkit* that allows the generation of ontology specific editor applications for topic map ontologies in a flexible multi-step process. The whole wizard-driven process is based on an ontology that is specified by the user as a Topic Maps schema and leads the user to the finished editor desktop application.

The Editor Generator Toolkit is an extension for the well-known *Eclipse IDE*, which not only provides the environment to develop applications in many programming languages, but also provides a basic architecture for standalone desktop applications. This architecture, the so-called *Eclipse RCP*, is used by the Editor Generator Toolkit as the basic application framework and is extended by an ontology specific domain model.

Section III contains an overview of the Editor Generator Toolkit architecture and its individual parts. Section IV describes the workflow that leads from the initial Topic Maps schema to the finished editor application. In Section V we introduce the Yacca editor as an example application. Section VI summarizes our results and provides an outlook on future work.

II. State of the Art

The tool we present here is no ontology editor in terms of products like Protégé or OntoStudio. Whereas these products offer features for creation and editing of knowledge bases with one of the popular ontology languages as OWL or RDFS, we present a toolchain to create an instance editor as a desktop application for a specific Topic Maps schema ontology [1]. However, an ontology editor is required to create the basic Topic Maps schema from which the editor application is built. This can be done with another Eclipse extension, called *Onotoa* [2], [3].

All currently existing Topic Maps domain editors like Ontopoly or Topincs are web-based applications and thus require both a central application server and a reliable network connection. As these requirements are not always available, we focus on desktop applications to avoid any obstacles for the users of the editor application.

Furthermore, we separate the process of ontology creation from the process of editing the actual data, so that users without knowlegde of the Topic Maps schema language are able to use the editor application that was previously created by a topic maps expert.

III. Architecture

The editor applications that are provided through the Editor Generator Toolkit consist of several components and are based on the Eclipse RCP. See Figure 1 for an overview of these components. The individual components, which were developed at the Topic Maps Lab, will be explained in this section.

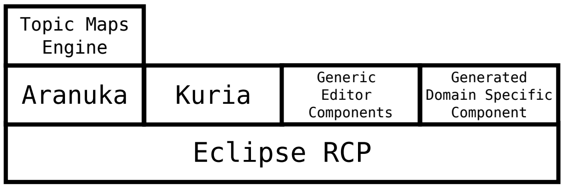

Topic Maps Engine
Aranuka
Kuria
Generic Editor Components
Generated Domain Specific Component
Eclipse RCP

Figure 1. The components of a generated editor application. Based on the Eclipse RCP, Aranuka handles the mapping of the domain model to the Topic Map engine while Kuria, the generic editor components, and the generated domain specific component provide the visible parts of the editor application.

A. Eclipse RCP

Eclipse is a plug-in based software development system which initially was created as integrated development environment (IDE) for Java. The environment is enhanced regularly by the work of the large community that grew around Eclipse. With version 3.0, the Eclipse Foundation released the Eclipse RCP (Rich Client Platform), which consists of a stripped version of the IDE and can be used to create new applications that make use of the architecture of Eclipse. The Eclipse RCP components form the foundation of the editor applications that are generated with the Editor Generator Toolkit.

Using the Eclipse RCP also results in platform independent applications. The *Standard Widget Toolkit* (SWT) is a Eclipse plugin and responsible for rendering the user interface. It is directly connected to the platform specific user interface libraries. The Eclipse Foundation provides a set of bindings for a lot of platforms, which can be used to create Eclipse RCP applications for different operating systems and hardware architectures. A complete description how to develop Eclipse RCP applications can be found in [4].

B. Aranuka

Semantic technologies like Topic Maps provide a generic way to represent data. On top of that, programming interfaces of Topic Maps engines like the *Topic Maps Application Programming Interface* (TMAPI) [5] need to be generic too and therefore developers must have a good understanding of Topic Maps even if they are supposed to write a domain specific application.

Aranuka, an open source project from the Topic Maps Lab, provides a way to map a domain model that was developed in Java to an underlying Topic Maps engine [6]. The developer then uses his model classes and a helper class from Aranuka called *Session*. This Session is able to retrieve topics from the topic map and persist topics back into the topic map. Aranuka uses connectors to associate Aranuka to a specific Topic Maps engine. These connectors bind the Aranuka core to any TMAPI supporting Topic Maps engine. Right now, Aranuka was tested with the Topic Maps engines *tinyTiM* and *Ontopia* and works flawlessly with both of them.

The configuration of the domain model is done via Java annotations. For every construct of the Topic Maps Data Model (TMDM) an specific annotation can be used in classes or attributes. Every annotation contains several properties which can be used to configure the mapping. The annotation **@Occurrence** for instance, has a type property that specifies the subject identifier of the occurrence type of the mapped attribute. More information about Aranuka annotations and its use can be found in [7].

After annotating the model classes a Configurator instance is used to tell Aranuka which classes should be mapped to topics and which connector should be used. It is also possible to add names to the types specified in the annotations. This is done via an internal mapping between the subject identifier set of the annotation and the value of the name that should be added to the used topic type. The Configurator provides additional methods for adding and removing prefixes, too. These prefixes can be used in any URI which is used as identifier in annotations and instances. Every instance which is mapped to a topic must have at least one identifier which should be a URI.

C. Kuria

The W3C created a group to analyze the need of model-based user interfaces (see [8]). In its report the groud states that the development of web applications should use utilities to build the final user interface based on the model of the application and the target platform. Instead of developing for different platforms, a description language based on XML should be used to map the domain model to specific user interface elements. With this description different layouts and designs can be generated, based on the target platform, which could be a mobile device or a standard browser.

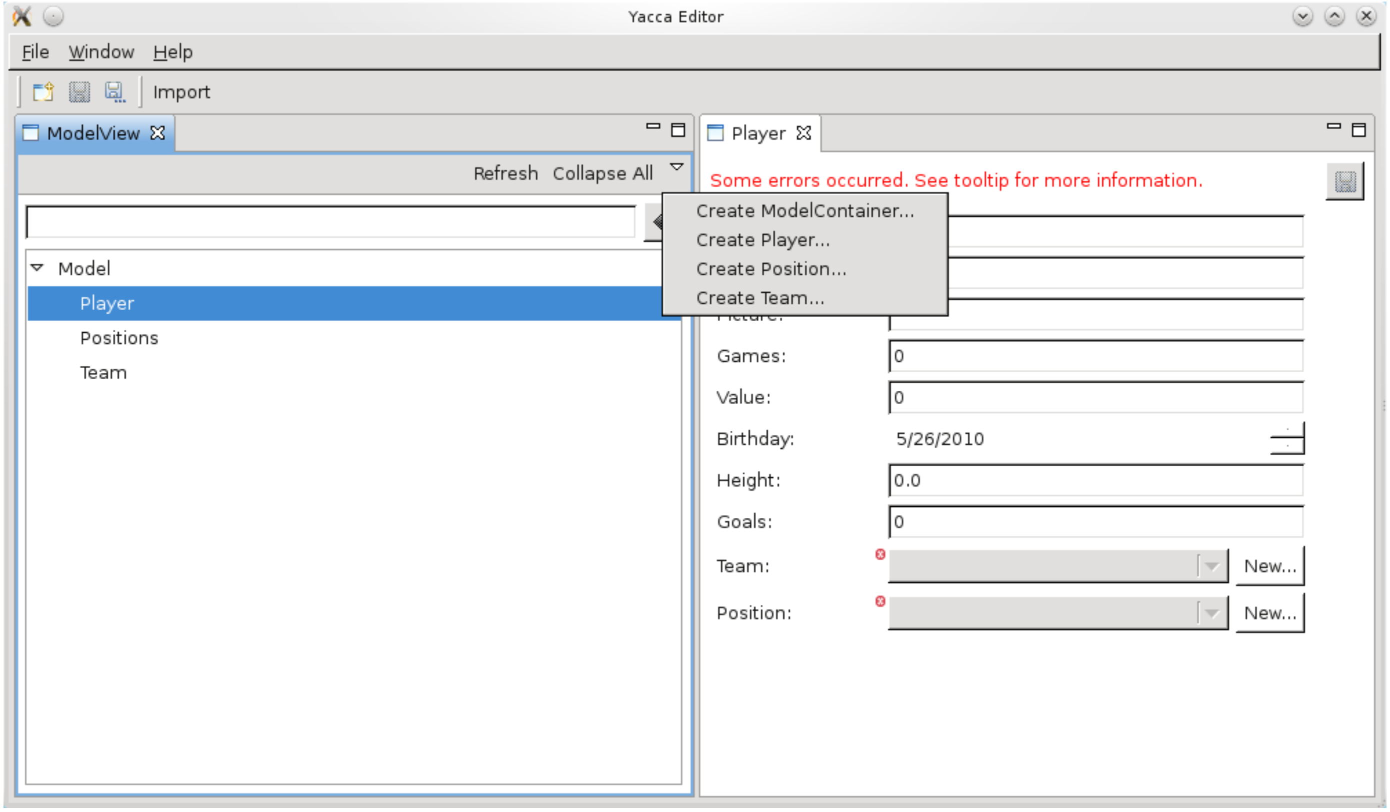
A similar approach is implemented with *Kuria*, an open source project from the Topic Maps Lab [9]. Instead of web applications, Kuria generates input masks for desktop applications. It is modularized to support different approaches of declaration and generation. The Editor Generator Toolkit uses three Kuria modules, the *Kuria Runtime* module, the *Kuria Annotation* module, and the *Kuria SWTGenerator* module.

Kuria Runtime: The Kuria Runtime module is the core of Kuria. User interfaces are composed of elements like buttons, windows, labels, dialogs, which are called widgets. For every widget exists a descriptor, which is called binding. Bindings contain the model specific information of the widget, for instance which text is valid for a text field and an accessor and mutator method. With the binding it is possible to set the value of an attribute of an object instance. In addition, bindings for tree nodes and table columns exists.

Kuria Annotation: The Kuria Annotation module is used to create widget bindings based on annotations on the model classes. If no annotation exists, a binding based on the datatype and the name of the field is created. It is also possible to hide an attribute, which can be done with the annotation **@Hidden**. For a complete list of annotations and their attributes refere to [10].

Kuria SWTGenerator: The *Standard Widget Toolkit* (SWT) was developed by IBM to create an efficient Java widget toolkit which uses the libraries of the underlying operating system. One other well known user interface (UI) library for Java is Swing, which is part of the Java SDK. Swing renders UI elements by itself, which results in a consistent look and feel, because applications using Swing look very much the same on every system. However, these applications look kind of alien in most operating systems.

In contrast, the SWT wraps UI elements of the operating system, and thus all applications that rely on SWT need platform dependent libraries for every system. Though this has the advantage of providing the look and feel of the



Some errors occurred. See tooltip for more information.

Games: 0

Value: 0

Birthday: 5/26/2010

Height: 0.0

Goals: 0

Team:

Position:

Figure 2. Empty editor window. On the *left side* is the modelview. On the *right side*, for every opened model instance an editor tab is opened.

underlying operating system. Another advantage is the increased speed that comes while using the operating system's native libraries, which is much faster than using the emulated ones, like Swing does.

The SWTGenerator module is used to generate a user interface based on the widget bindings using SWT. In addition, it provides methods to easyly generate tree and table widgets. To create an input mask for an instance, the SWTGenerator uses the parent widget and the class of the instance. The SWTGenerator then checks whether a binding for the class exists and generates the input mask according to the bindings.

The Editor Generator Toolkit produces Kuria annotations for every generated domain model. In the resulting editor application, the overview tree and all input masks are generated by Kuria.

D. Generic Editor Components

The *Generic Editor Components* provide the containers for the user interfaces. These components are fixed and configured by the generated annotated domain model.

In Figure 2, an empty application window is shown. On the left side is the *ModelView*. This window contains a tree representing the model. For every type in the ontology a node exists and its children are the instances of the type. New instance of a model type can be created with the provided context menu. Alternatively, the view provides a menu on the left side of the title bar with options for every topic type. Already existing instances can be edited by simply clicking on them.

The individual editors for the instances are placed on the right side. It is possible to open multiple editors. If an editor is activated the *Save*-button in the toolbar persists the edited instance into the topic map. Every editor provides a *Save and Close* button inside the editor window which persists the instance in the topic map and closes the editor.

Generated Domain Specific Component: The last component of the Editor Generator Toolkit (cf. Figure 1) is the *Generated Domain Specific Component*. This plug-in contains the generated domain model classes and additional product configurations. The latter are necessary for the configuration of the *Eclipse IDE* and contain information about the plug-ins that are part of the editor and thus must be exported together with the editor application. The configuration of Aranuka is also part of this plug-in and can be found in the **ModelHandler** class. After generating the code, the names for types can be added there. The selection of the Aranuka connector is also possible in this class and can be changed any time after the generation process.

IV. Generate an Editor

This chapter explains how to generate an editor application, what prerequisites are needed and what steps are necessary to create the application. An overview of the steps are shown in Figure 3.

Prerequisites

The Editor Generator Toolkit consists of a set of Eclipse plug-ins that add the generation facility to the Eclipse IDE. In order to work with these plug-ins, a working Eclipse with the *Java Development Toolkit* (JDT) is needed. To create the base ontology, it is strongly recommended to use the Topic Maps schema editor Onotoa.

Create the Ontology

The first step is to create the ontology. This can be done via text editors writing plain TMCL in CTM-notation or any visual editor. We advise to use Onotoa, a visual editor which can be installed directly into the Eclipse development environment.

After creating the model in Onotoa it should be exported to TMCL which is indicated by step 2 in the activity diagram in Figure 3.

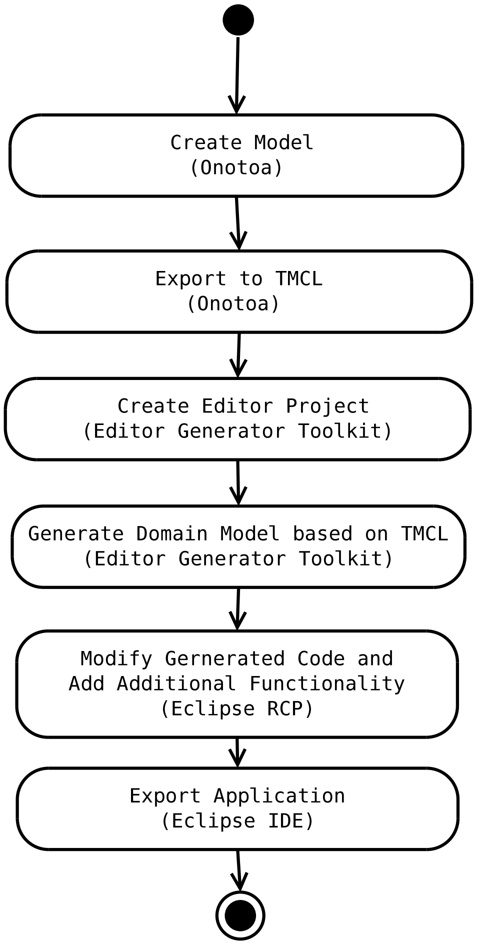

Create Model
(Onotoa)
Export to TMCL
(Onotoa)
Create Editor Project
(Editor Generator Toolkit)
Generate Domain Model based on TMCL
(Editor Generator Toolkit)
Modify Gernerated Code and
Add Additional Functionality
(Eclipse RCP)
Export Application
(Eclipse IDE)

Figure 3. Chain of activities to develop generate an ontology specific editor.

Create an Editor Project

The Editor Generator Toolkit adds a new wizard to the *New Project Wizard* list of Eclipse. Create a new project and fill in all required data into the first page of the wizard. This page asks for a project name, which should have a form like a Java package name. The name of the application will be used in the title bar of the application. It will be used as name of the executable binary of the application, too. The third entry is a drop box which allows selection of the used Topic Maps engine.

In the second editor page the name of the schema file is required. If this field is empty no model will be created and the developer has to create it on his own.

Modify Generated Code and Add Additional Functionality

The generated domain model is a set of Java classes which are generated on the basis of the topic types in the given TMCL schema. In addition, these classes have attributes annotated for Kuria and Aranuka. The generated classes can be revised and modified to tailor the editor and receive the expected input masks for the models. Examples for modifications are:

- A topic has an occurrence of type string. In the class an attribute is generated with a **@Textfield** annotation of Kuria. This annotation indicates the use of a text field with one line. To have a multiline text area an additional attribute must be added to the annotation.
- For topics with associations in every generated class an attribute for the counter player exists. This is because of the bidirectional nature of associations in topic maps. In most cases, only one direction is needed in the editor interface. It is recommended to remove one of the counter player attributes.

The generated editor is an eclipse application and therefore provides some possibilities to extend the application. These can be done inside the generated domain specific plug-in or in additional plug-ins. All new dependencies should be added to the product configuration, which is responsible for the correct export of every required plug-in. Developing additional functionality is optional and not needed to export a working editor application.

Export the Application

The editor for the product configuration provides an export function, which is a link in the first page of the editor. By activating the link, a wizard opens and asks for the target directory and the desired target platforms.

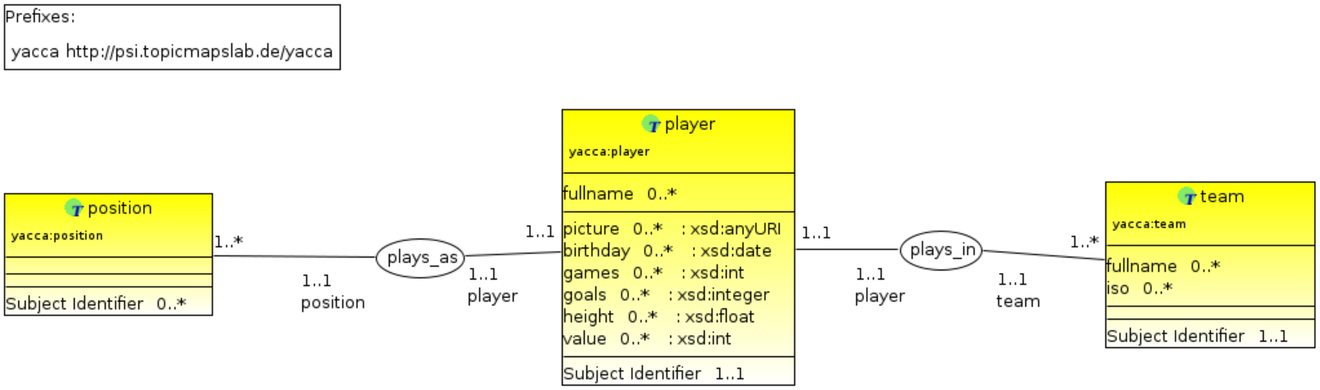


Figure 4. The Yacca ontology as it was created with Onotoa.

After the export is finished, the target directory contains the created applications, one for each platform, which can then be deployed to the target machines.

V. Example Editor: Yacca

In this section, we present an example on how a generated editor application can be used in the developmental process of Topic Maps driven web applications. The example we provide is called *Yacca* and was voted into top ten at the 2010 ESWC AI Mashup challenge.

In Yacca, the power of structured data – that comes with the use of Topic Maps – as data store is combined with the flexibility of TMQL as a query language that allows to extract just the desired amount of data from the Topic Map. The so called ”Yacca cards“ are HTML snippets, produced from the structured data and provided by *Maiana*, the social Topic Maps explorer from the Topic Maps Lab [11]. The topic map for the data store was created with an editor that was generated with the Editor Generator Toolkit.

The Yacca topic map contains three topic types and two associations (cf. Figure 4). Based on this Topic Map schema, the editor in Figure 5 was created. The ModelView shows the three topic types that are now classes in the Java application. On the right side, an editor for a player is visible. The dialog on the left contains another input mask for the position topic. This dialog opens when pushing the *New* button next to the position field in the editor. A similar dialog opens for the team of the player.

The Yacca editor has an additional feature: An import of players from a comma separated file. This function was used after creating the team and position topics.

After creating the topic map with the editor, it was uploaded to Maiana [12]. The editor is still used for updating the topic map, for instance in case a player gets injured and can not play.

VI. Conclusion and Future Work

Summary

The Editor Generator Toolkit provides an simple and fast way to generate small and fast Topic Maps editor applications that are feasible for easy collation of data. Based on the popular Eclipse Platform it integrates into most established development processes. With the used base technologies – Eclipse RCP and the Standard Widget Toolkit – the generated editor can be used on almost every platform. With our approach, Topic Maps experts with little to no experience in Java programming are able to build editor applications based on their Topic Maps schema.

In this paper, we have explained the architecture of the Editor Generator Toolkit and its base modules. Furthermore, we showed the process of creating the editor application and have given an example for successful use of a generated editor.

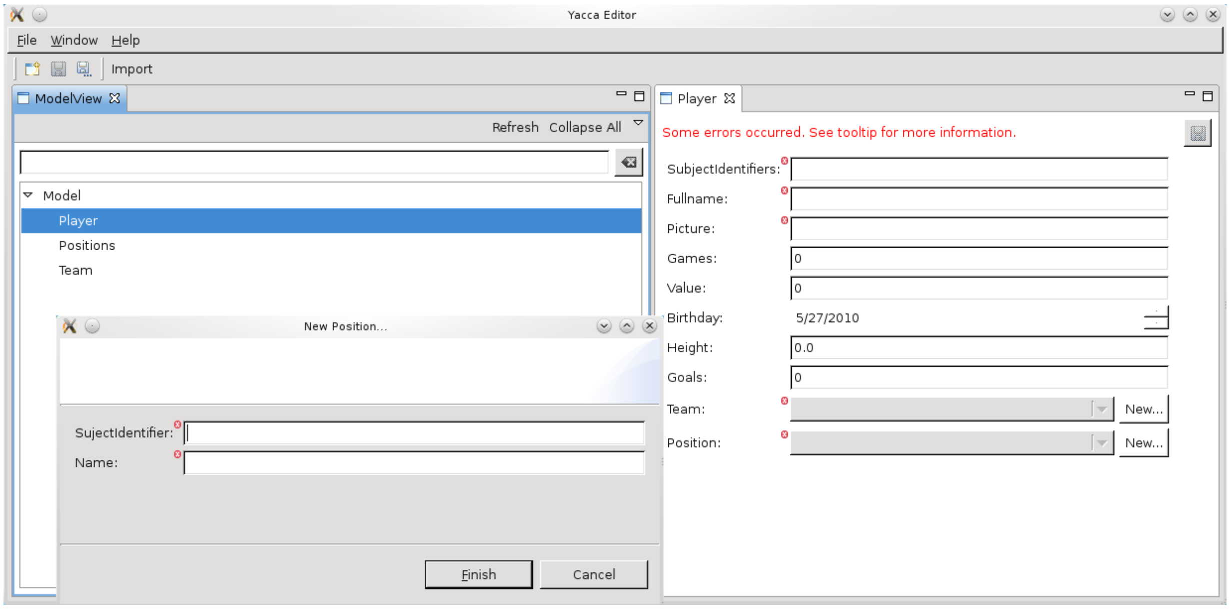
Future Work

The generated editor is an application with a simple user interface. Especially with a lot of topics the tree in the ModelView gets to large. Search facilities can be implemented to provide ways to find topics with a specific property. This could be done by specifying the property inside a dialog or entering a TMQL query [13].

In the current state, the generator produces default Kuria annotations and attributes for every site of an association. In future release some additional schema elements in form of reification or occurrences of specific types should be introduces and supported by the schema editor. With these additional elements the manual modification of the generated code would be unnecessary.

References

[1] M. Weiten, *Semantic Knowledge Management*. Springer, 2009, ch. OntoSTUDIO as a Ontology Engineering Environment, pp. 51–60.



Some errors occurred. See tooltip for more information.

SubjectIdentifiers:
Fullname:
Picture:
Games: 0
Value: 0
Birthday: 5/27/2010
Height: 0.0
Goals: 0
Team:
Position:

SujectIdentifier:
Name:

Figure 5. The Yacca editor application with ModelView, Editor and Dialog to create a new position.

[2] H. Niederhausen, “Onotoa – TMCL-basierter grafischer Schema-Editor für Topic Maps,” Master’s thesis, University of Leipzig, 2009.

[3] H. Niederhausen, *Onotoa Handbook*, last checked: July 19 2010. [Online]. Available: http://docs.topicmapslab.de/onotoa

[4] J. McAffer and J.-M. Lemieux, *Eclipse Rich Client Platform*. Addison-Wesley Professional, 2005.

[5] L. Heuer and J. Schmidt, “Tmapi 2.0,” in *TMRA – Subject Centric Computing*, 2008, p. 129.

[6] H. Niederhausen, *Aranuka Project Site*, last checked: July 19 2010. [Online]. Available: http://code.google.com/p/aranuka

[7] H. Niederhausen, “Aranuka documentation,” last checked: July 19 2010. [Online]. Available: http://docs.topicmapslab.de/aranuka

[8] J. M. C. Fonseca, “W3C Incubator Group Report 04 May 2010,” 2010, last checked: July 19 2010. [Online]. Available: http://www.w3.org/2005/Incubator/model-based-ui/XGR-mbui-20100504/

[9] H. Niederhausen, *Kuria Project Site*, last checked: July 19 2010. [Online]. Available: http://code.google.com/p/kuria

[10] H. Niederhausen, “Kuria Documentation,” last checked: July 19 2010. [Online]. Available: http://docs.topicmapslab.de/kuria

[11] Topic Maps Lab, *Maiana*, last checked: July 19 2010. [Online]. Available: http://maiana.topicmapslab.de

[12] Sven Windisch, *Yacca Topic Map*, last checked: July 19 2010. [Online]. Available: http://maiana.topicmapslab.de/u/yacca/tm/yacca

[13] L. M. Garshol and R. Barta, “Topic maps query language,” last checked: July 19 2010. [Online]. Available: http://www.isotopicmaps.org/tmql/tmql.html

Using WordNet for Concept-Based Document Indexing in Information Retrieval

Fatiha Boubekeur
Department of Computer Sciences,
Mouloud Mammeri University of Tizi-Ouzou
Algeria
amirouchefatiha@mail.ummto.dz

Mohand Boughanem, Lynda Tamine, Mariam Daoud
IRIT-SIG,
Paul Sabatier University of Toulouse III,
France
{boughane, tamine, daoud }@irit.fr

***Abstract*—Concept-based document indexing deals with representing documents by means of semantic entities, the concepts, rather than lexical entities, the keywords. In this paper we propose an approach for concept-based document representation and weighting. Particularly, we propose (1) an approach for concept-identification (2) and a novel concept weighting scheme. The concepts are first extracted from WordNet and then weighted by means of a new measure of their importance in the document. Our conceptual indexing approach outperforms better than classical keyword-based approaches, and preliminary tests with the weighting scheme give better results than the classical tf-idf approach.**

Keywords-Information retrieval; conceptual indexing; concept weighting; WordNet.

I. INTRODUCTION

Information retrieval (IR) is concerned with selecting from a collection of documents those that are likely to be relevant to a user information need expressed using a query. Two basic functions are carried out in an information retrieval system (IRS): document indexing and query-document matching. The main objective of indexing is to assign to each document (respectively query) a descriptor represented with a set of features, usually weighted keywords, derived from the document (respectively query) content. The main goal of query-document matching, also called query evaluation, is to estimate the relevance of a document with respect to the query. A key characteristic of classical IR models is that the degree of query-document matching depends on the number of the shared terms. This leads to critical problems induced by disparity and ambiguity.

- Disparity refers to the property that has some terms to be represented by different words and associated to identical or related senses. Disparity causes relevant document to not be retrieved. For example, a document on *unix*, nevertheless relevant for a query on *operating systems*, will not be retrieved if the words *operating* and *systems* are absent in this document.
- Ambiguity refers to two properties: homonymy and polysemy [14]. Homonymy refers the property that has some terms, represented by the same word, to be associated to different meanings. The *bark* of a dog versus the *bark* of a tree is an example of homonymy. Polysemy is related to the property of some words to express different meanings. *Opening* a door versus *opening* a book is an example of polysemy. In classic IRS, ambiguity causes irrelevant documents to be retrieved. For example, a document on *politics in France*, nevertheless not relevant for a query on *Anatole France*, will be retrieved because of the shared word *France*. Various approaches and techniques have attempted to tackle these problems by enhancing the document representation or query formulation. Attempts in document representation improvements are related to the use of semantics in the indexing process. Semantic indexing aims at representing documents (and queries) by means of senses (concepts) rather than simple words. Senses are identified (ie. disambiguated) by means of *word sense disambiguation* (WSD) approaches that allow finding the right sense of a word in a given context. WSD are classified in supervised and unsupervised approaches [32]:
- Supervised WSD uses training Corpora [8][15][19] to first build the required knowledge base for disambiguating senses. The related approach consists on examining a number of contexts of the target word (that is the word to disambiguate), in a training corpus, from which rules on word arrangement (co-occurrence, ordering, contiguity) [29], or word usage [24] are constructed. This knowledge is then used for further recognition of word sense in a given context.
- Unsupervised WSD use external linguistic resources such as MRD (*Machine Readable Dictionnary*) [11] [16][27][30], thesaurus [31], ontologies [21][25] or Wikipedia [18] in order to identify word senses instead of using “trained” senses. This is called conceptual (or concept-based) indexing.

In this paper, we propose a conceptual indexing approach based on the use of a linguistic resource namely WordNet. The main idea of our approach is to classify document words into WordNet entries, then to associate them with correct senses. We propose to use WordNet [20] as source of evidence for word sense identification and for sense weighting.

The paper is structured as follows: Section II introduces the problems of semantic indexing and then reports some related works and presents our motivations. In Section III, we detail our proposed semantic indexing approach.

Preliminary experimental results are presented in Section IV. Section V concludes the paper.

II. Related work and motivations

A. The Problem

Conceptual indexing approaches generally rely on deriving concepts from linguistic resources such as MRD, thesaurus, and ontologies in order to identify the relevant sense (concept) of a word in a given context. For this aim, the indexing process poses two key problems: concept identification and concept weighting.

- Concept identification aims at assigning mono-words or multi-words to the most accurate entries in the ontology. Identifying representative words is a classical indexing problem. Classical approaches are based on linguistic (tokenization, lemmatization, stop-words eliminating) and statistical techniques to identify keywords in the document. Given these keywords, a key problem in semantic indexing is to identify for each keyword its right sense(s) in the document. This leads to a WSD problem.
- Concept weighting. The purpose of concept weighting is to quantify the degree of importance of each concept in the document. Weighting is a crucial problem in IR. Indeed, the quality of retrieving depends on the quality of weighting. Good weighting is required to guarantee that the relevant documents are retrieved for a given query. In classical IRS, the well known *tf*idf* weighting scheme is successfully used. In the context of conceptual indexing, the challenge is how to correctly weight concepts.

In what follows, we give an overview of the WordNet structure, a survey of related works and then highlight the key points of our approach.

B. WordNet Overview

WordNet is an electronic lexical database [20] which covers the majority of names, verbs, adjectives and adverbs of the English language, which are structured in a network of nodes and links.

1) *Nodes:* also called synsets are sets of synonyms.

- A synset is a concept.
- A concept, which is a semantic entity, is lexically represented by a term.
- A term is a word (mono-word term) or a group of words (multi-word term) that represents a concept.

2) *Links:* Links represent semantic relations between synsets, in which the hypernym-hyponym relations defined as follows:

- the *is-a* relation (also called *subsumption* relation) associates a general concept (the hypernym) to a more specific one (its hyponym). For example, the name *tower#*1[1] has as hyponyms *silo, minaret, pylo…* The *is-a* relation thus organizes WordNet synsets into a hierarchy of concepts. An example of hierarchy of synsets corresponding to the word "*dog*" is given in Table 1.
- the *instance* relation links a concept (hypernym) with its instance (hyponym). For example, the name *tower#*1 has for instance "*Eiffel tower*".

TABLE I. WORDNET SYNSETS OF THE WORD "DOG"

Noun
S: (n) dog, domestic dog, Canis familiaris (a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds) "the dog barked all night"
S: (n) frump, dog (a dull unattractive unpleasant girl or woman) "she got a reputation as a frump"; "she's a real dog"
S: (n) dog (informal term for a man) "you lucky dog"
S: (n) cad, bounder, blackguard, dog, hound, heel (someone who is morally reprehensible) "you dirty dog"
S: (n) frank, frankfurter, hotdog, hot dog, dog, wiener, wienerwurst, weenie (a smooth-textured sausage of minced beef or pork usually smoked; often served on a bread roll)
S: (n) pawl, detent, click, dog (a hinged catch that fits into a notch of a ratchet to move a wheel forward or prevent it from moving backward)
S: (n) andiron, firedog, dog, dog-iron (metal supports for logs in a fireplace) "the andirons were too hot to touch"
Verb
S: (v) chase, chase after, trail, tail, tag, give chase, dog, go after, track (go after with the intent to catch) "The policeman chased the mugger down the alley"; "the dog chased the rabbit"

C. Related Work

Conceptual indexing approaches represent documents by concepts. These concepts are extracted from ontologies and other linguistic resources. The indexing process generally runs in three steps: (1) keyword extraction, (2) sense identification and (3) concept weighting.

1) *Keyword extraction:* keywords are extracted from the document by a classical indexing approach (tokenisation, elimination of empty words, then lemmatization) [1][2][3][4][13][26][28]. Keywords are then mapped on the ontology in order to identify the corresponding concepts (or sense). As an ambiguous term may correspond to several entries (sense) in the ontology, it is must be disambiguated. To disambiguate a word sense, Voorhees [28], classifies every synset of this word on the basis of the number of words collocated between a neighborhood of this synset and the local context (the sentence in which the word occurs) of the corresponding ambiguous word. The best classified synset is then considered as the adequate sense of the ambiguous word. In a similar approach, Katz et al [26] define the local context of a word as the ordered list of words starting from the closest useful word to the left or right neighborhood until the target word. To disambiguate word sense, Katz et al. first extract words (called *selectors*) from the local context of the target word. Then the set *S* of selectors is compared with the synsets of WordNet. The synset that has the maximum words in common with *S* is selected as the adequate sense of the target word. To

[1] *tower#1* refers to the first sense of the word *tower* in wordNet.

disambiguate an ambiguous word, Khan et al. [13], proposed an approach based on the semantic closeness of concepts. The semantic closeness of two concepts is calculated by a score based on their mutual minimal distance in the ontology. The concepts that have the highest scores are then selected. Based on the principle that, among the various possible senses (candidate senses) of a word, the most adequate one maximises its relations with other document word candidate senses, Baziz et al. [1], assign a score to every candidate sense (candidate concept) of a given word in the document. The score of a candidate concept is obtained by adding its semantic relatedness values [16][17][22] with other candidate concepts in the document. The candidate concept having the highest score is then selected as the adequate sense (concept-sense) of the associated index word. In our approach proposed in [3][4] this score is based on the sum of its similarity value with other candidate concepts in the document, balanced by their respective frequencies.

2) Concept weighting: Analogously to term weighting in classical keyword-based IRS, weighting concepts aims at assigning to each concept its importance in a document. Weighting concepts approaches decline in two main tendencies: (1) lexical weighting (2) semantic weighting approaches.

In lexical weighting approaches, the lexical concept weighting, concepts are considered through the terms which represent them. Hence, concept weighting consists on term weighting. The weighting approaches of Baziz et al [1] and Voorhees [28] relie on this principle. Based on the extended vectorial model introduced in [10], in which every vector consists of a set of sub-vectors of various concept types (called *ctypes*), Voorhees [28] proposed to weight concepts by using a normalized classic *tf***idf* scheme. The approach proposed by Baziz et al. [1], extends the *tf***idf* scheme to take into account compound terms. The proposed approach, called *Cf***idf*, allows to weight simple terms and compound terms associated with concepts. Indeed, the weight of a term is based on the cumulative frequency of the term itself and of its components.

While in the semantic concept weighting approaches, concepts are considered through their senses. Concept weighting approaches aim at evaluating the importance of the senses in a document content. This importance is estimated through the number of semantic relations the concept has with other concepts in a document. The approaches proposed in [5][9][12] are based on this principle. In addition to the concept weighting, semantic relations are also weighted in [12]. In the same context, in the approach proposed by Boughanem et al. [5], the number of relations of a concept with the other concepts in the document defines a measure called centrality of the concept. The authors combine centrality and specificity to estimate the importance of the concepts of a document. The specificity of a concept is its depth in the WordNet hierarchy. In our work introduced in [3][4], we focused on combining both semantic and lexical concept weighting. Indeed, we propose to weight compound terms (representing concepts) on the basis of a probabilistic measure of senses relatedness between terms and associated sub-terms and sur-terms. Practically, the weight of a given term *t* is based on a probabilistic measure of the possible senses of term *t* (noted *Sens(t)*) relatively to the senses of its sub-terms (*Sub(t)*) and its sur-terms (*Sur(t)*) [3] taking into account their respective frequencies in the document. The probability that a term *t* is a possible sense of a term *t'* is measured as the fraction of the number of *t's* senses including term *t'*, over the number of all senses of term *t*.

Formally:

$$P\left(t \in Sens(t')\right) = \frac{|\left\{C \in Sens(t') / t \in C\right\}|}{|Sens(t')|}. \quad (1)$$

$$W_{t,d} = \left(\begin{array}{l} tf(t) \\ + \sum_i tf(Sur_i(t)) \\ + \sum_j \left[P\left(t \in S\left(Sub_j(t)\right)\right) * tf\left(Sub_j(t)\right) \right] \end{array} \right) * \ln\left(\frac{N}{df(t)}\right). \quad (2)$$

Where *N* represents the total number of documents in the corpus and *df(t)* the document frequency.

D. Our Contribution

Our approach proposed in this paper is a revisited version of the theoretic framework proposed in [3][4]. The key objective of our approach is to represent the document by a semantic kernel, composed of weighted concepts extracted from WordNet. In this paper, we redefine the approach of concept identification and concept weighting as follows:

1) The proposed approach of concept identification in this paper is based on the overlapping degree between a WordNet synset and the local context (the sentence) in which the word appears in the document. Unlike the approach proposed in [3], this approach presents the advantage to allow the detection of collocation of words independently of their order of appearance in the context.

2) The weighting approach proposed in this paper is based on a new measure of concept importance in a document. This measure takes into account semantic relatedness between concepts on one hand, and the concept frequency in the document on the other hand. The concept frequency is revisited so as to take into account multi-word representations of a concept.

III. OUR CONCEPTUAL DOCUMENT INDEXING APPROACH

We propose to use WordNet to build the document representative semantic index. The document indexing process is handled through three main steps: (1) Identifying WordNet concepts, (2) Assigning concepts to document index terms and (3) Weighting concepts. In the following, we present these steps.

A. Concept Identification

The purpose of this step is to identify WordNet concepts that correspond to document words. Concept identification is based on the overlap of the local context of the analyzed word with every corresponding WordNet entry. The entry which maximizes the overlap is selected as a possible sense of the analyzed word. The concept identification algorithm is given in Table 2.

TABLE II. CONCEPT IDENTIFICATION ALGORITHM

Input: document *d*

Output: $N(d)$, the set of all WordNet concepts belonging to terms (words or word- collocations) in *d.*

Procedure:

Let w_i be the next word (assumed not to be a stop word), to analyze in the document *d*. We define ψ_i the context of word in the document as the sentence in *d* that contains the word occurrence being analyzed:

1. Compute $\zeta_i = \{C_1, C_2, ..., C_n\}$ the of WordNet entries containing w_i. Each $C_j \in \zeta_i$ is represented by a multi-word or mono-word term.
2. ζ_i is ranked as follows: $\zeta_i = \{C_{(1)}, C_{(2)}, ..., C_{(n)}\}$ where $j = (1), ..., (n)$ is an index permutation such as $\left|C_{(1)}\right| \geq \left|C_{(2)}\right| \geq ... \geq \left|C_{(n)}\right|$, where $\left|\ \right|$ denotes the concept length, in terms of number of words in the corresponding terms..
3. For each element $C_{(j)}$ in ζ_i, do:

- Compute the intersection $\eta = \cap(\psi_i, C_{(j)})$ as the set of common words between ψ_i and the representative term of C_j.
- If $|\eta| < \left|C_{(j)}\right|$ then the concept-sense $C_{(j)}$ is not within the context ψ_i
- If $|\eta| = \left|C_{(j)}\right|$ then the concept-sense $C_{(j)}$ is within the context ψ_i. $C_{(j)}$ is added to the set of possible senses associated with the document

4. The process is repeated for each concept sense C in ζ_i, for which $|C| = \left|C_{(j)}\right|$.

B. Term Disambiguation

Each term t_i in document d may be associated to a number of related possible senses ("i.e." WordNet concepts) S_i. To disambiguate a term t_i, we associate a score to each of its possible senses, based on its semantic relatedness to other concepts in *N(d)*. The concept C_i which maximizes the score is then selected as the best sense of term t_i.

Formally:

$$C_i = \underset{C_i \in S_i}{\arg\max} \left(\sum_{\substack{à \leq j \leq |N(d)| \\ j \neq i}} \sum_{c_k \in S_j} occ(C_i) * occ(C_k) * Dist(C_i, C_k) \right) \quad (3)$$

Where *occ(C$_i$)* is the number of *C$_i$'s* occurrences in the document, and *Dist(C$_i$, C$_k$)* is the semantic relatedness between concepts C_i and C_k.

The set of all selected senses represents the semantic core of the document *d*.

C. Concept Weighting

Our objective here is to assign to each concept in *N(d)*, a weight that expresses its importance. For this aim, we first introduce some definitions and then present our concept weighting approach.

1) Definitions: Let *C* and *C'* be two concepts in *N(d). C* and *C'* are represented by terms *t* and *t'* respectively.

Definition 1: *t'* is a sub-term of *t,* if the set of words that compose *t* includes the set of words that compose *t'*.

Definition 2: *C'* is a sub-concept of *C,* if *t'* is a sub-term of *t.*

Let $Sub_j(C)$ be the set of all sub-concepts of concept *C*. We note $Sens(C)$ the set of all WordNet senses semantically related to *C*.

Definition 3: *C'* is a possible sense of *C,* if $C' \in Sens(C)$.

2) The Weighting approach: Our concept weighting approach is based on the following assumptions:

- the more a concept is frequent and strongly correlated to other concepts in the document, the more it is important,
- The frequency of a concept relies on its occurrences and the occurrences of its sub-concepts in the document.

Based on these assumptions, we propose a concept weight scheme based on:

- The semantic relatedness, $Dist(C_i, C_j)$, between the considered concept C_i and other concepts C_j in *N(d)*.
- The frequencies of the related concepts. The frequency of a given concept *C* depends on its own occurrences in the document, and on the occurrences of its sub-concepts $Sub_j(C) \in N(d)$, balanced by the probability that the sub-concept expresses a related meaning to the concept.

Formally:

$$W(C_i) = \sum_{i \neq j, 0 \leq i, j \leq |N(d)|} tf(C_i) * tf(C_j) * Dist(C_i, C_j). \quad (4)$$

And:

$$tf(C_i) = occ(C_i) + \sum_{C_k \in Sub(C),} occ(C_k) * P(C_k \in (Sens(C_i))). \tag{5}$$

Where $P(C_k \in (Sens(C_i)))$ is the probability that C_k is a related sense of C_i.

Formally:

$$P(C_k \in (Sens(C_i))) = \frac{Dist(C_i, C_k)}{\max\limits_{C_j \in Sens(C_i)} (Dist(C_i, C_j))}. \tag{6}$$

IV. Experimental Evaluation

Our evaluation objective is to (1) measure the effectiveness of our proposed approach compared to classical indexing approaches and to (2) study the effect of concept weighting approach compared to classical term weighting.

In the following, we first present the experimental settings (the test collection and the evaluation protocol), then present and discuss the evaluation results of both our concept identification and concept weighting approaches.

1) The Test Collection: For our experiments, we used Muchmore test collection [7]. Muchmore is a parallel corpus of English-German scientific medical summaries obtained from the Web site of Springer. It declines in two versions among which an annotated one and a non annotated one. We used only the collection of non annotated English texts. This latter consists of 7823 documents and 25 queries. Relevant assessments are associated with each query.

2) Evaluation Protocol: The approach is evaluated using Mercure IR system [6]. The evaluation is made according to the TREC protocol. More precisely, every query is submitted to the system with the fixed parameters. The system returns the first 1000 documents for each query. The precision P5, P10, P20 and MAP (average precision) are computed. The precision *Px* at point *x* (*x*=5, 10, 20), is the ratio of the relevant documents among the first *x* returned documents. MAP is the mean average precision. We then compared the results obtained from our approach to different baselines.

A. Evaluation of Concept Identification Approach

Our objective of this experiment is to evaluate the impact of the semantic index quality on the retrieval effectiveness. For this aim, we compare two indexes:

- The first one is the semantic index composed of concepts, identified using our concept identification approach introduced in Section III, where each concept is weighted by means of *tf*. This approach is noted Concepts-TF in Figure 1.
- The second index is composed of a combination of both concepts and simple keywords weighted by means of *tf*. Keywords refer to those words that have no entries in WordNet. This approach is noted Concept-Fusion in Figure 1.

Retrieval results obtained using each of these two indexes are compared to two baselines:

- The first one is a classic baseline based on keyword-based indexing, where terms are weighted by means of classical *tf*idf* scheme. This approach is noted Classic-TFIDF in Figure 1.
- The second baseline is based on a keyword-based indexing where terms are weighted according to the BM25 scoring function [23].

Remark: No comparison was made with our approach proposed in [3][4], which mainly remains a theoretical framework. Indeed, this latter approch was not fully implemented (due to the complexity of its induced calculations), and only partial related results were available.

The evaluation results obtained for these different models are presented in Figure 1. According to the results, we conclude the following:

- Concepts-TF approach is better than the Classic-TFIDF baseline. The percentage of improvement is of 61 % for P5, 51 % for P10, 54 % for P20 and 51 % for the MAP
- The Concepts-Fusion approach is better than the Concepts-TF approach. The percentage of improvement is of 20 % for P5, 19 % for P10, 15 % for P20 and 23 % for the MAP. To study the statistical significance of these improvements, we have calculated the Wilcoxon signed-rank test between each indexing model and the baseline search performed by *tf*idf* weighting scheme. We assume that the difference between models is significant if the p-value p <0.1 and very significant if p<0.05. We have obtained a very significant p-value according to the Wilcoxon test of our model compared to classical indexing at almost the precision, P5, P10, P20 and MAP (see Table III). This proves the statistical significance of our indexing model to classical one. These results consolidate us in the idea that a combined indexing concepts+keywords is more effective than a concept-based indexing.

TABLE III. Statistical results from Wilcoxon test

p-value at	Classic-TFIDF vs. Concept-TF	Classic-TFIDF vs. Concept-Fusion
	P	P
P5	0,0015	< 0,0001
P10	0,0081	0,0002
P20	0,0042	0,0001
MAP	0,0102	< 0,0001

- Besides, our Concepts-Fusion approach presents better results than Classic-TF baseline with increasing rates of 94 % for P5, 45 % for P10, 77 % for P20 and 77 % for the MAP. Nevertheless, as shown on Figure 1, the Concepts-Fusion approach results are worse than those of the Classic-OKAPI baseline with decreasing rates of 0 % for P5, -1 % for P10, -5 % for P20 and -3 % for the MAP. This shortcoming is probably due to the imprecision of the disambiguation approach. Indeed, in a context of a precise disambiguation, we expect that indexing by the concepts will bring higher performance than indexing with keywords.

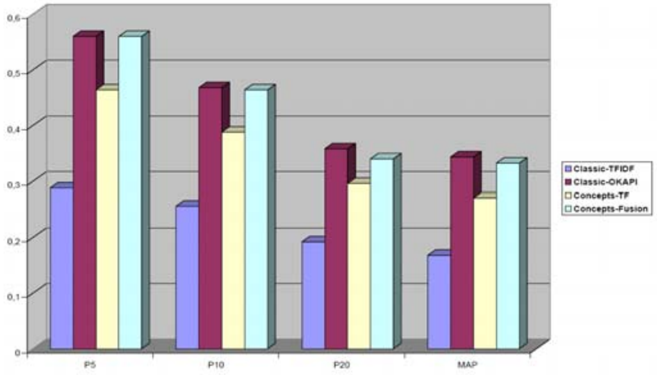


Figure 1. Concept vs keyword indexing.

B. *Evaluation of Concept Weighting Approach*

The second series of our experiments focuses on the evaluation of our concept-weighting approach introduced in Section II.C. Practically, we aim at measuring the impact of the weighting-scheme on the retrieval effectiveness. For this aim, we compare the effectiveness of two indexes:

- The first one consists on the concepts detected by our approach proposed in Section II.B, balanced by their respective frequencies. This approach is noted Concepts-TF in Figure 2.
- The second index consists on the concepts detected by our approach proposed in Section II.B, balanced by the proposed weight defined in Section II.C. This approach is noted Concepts-Score in Figure 2.

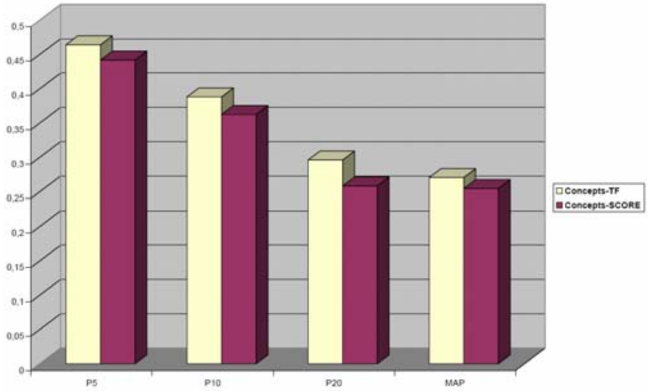


Figure 2. TF vs Score weighting in concept-based indexing.

Figure 2 presents a comparison between these two weighting approaches. From this figure, it appears that the results obtained from our proposed concept-weighting approach are globally less effective compared to those obtained from the frequency-based concept weighting scheme, with decreasing rates of -5 % for P5, -6 % for P10, -12 % for P20 and -6 % for the MAP. The obtained results are clearly below of our expectations. The problem behind this shortcoming improvement is probably due to the ranking score, used by Mercure search engine [6] to estimate the correspondence of a document to a query. Indeed, in evaluating the Concepts-Score index, instead of a *tf-idf* combination, the ranking score combines the concept weight with the non-correlated *idf* measure. This leads to decrease the precision improvement of the retrieved results.

V. CONCLUSION AND FUTURE WORK

We have presented in this paper a novel approach for conceptual document indexing. Our contribution concerns two main aspects. The first one consists on a concept-indexing approach based on the use of WordNet. The approach is not new but we proposed new techniques to identify concepts and to weight them. Preliminary results showed that our proposed concept-identification approach is more effective than a classical keyword-based indexing approach, and brings significant increasing rates compared to the Classic-TFIDF approach. However, this approach, even if combined with keywords, does not perform as well as the Classic-OKAPI baseline, probably due to the slight imprecision of our disambiguation. Besides, the concept-weighting approach produced reserved results. The likely cause of this unexpected shortcoming is the non-relevance of the ranking score for the semantic index. In future works, we plan first to revisit our concept disambiguation approach, and second to propose a ranking score for semantic indexes, which takes into account semantic weights of concepts. Works in this direction are in progress. .

ACKNOWLEDGMENTS

This work was made possible thanks to the financial support of the A.U.F. (*Agence Universitaire de la Francophonie*) and of U.M.M.T.O. (*Université Mouloud Mammeri de Tizi-Ouzou*) with the kind collaboration of IRIT (*Institut de Recherche en Informatique de Toulouse*).

REFERENCES

[1] M. Baziz, M. Boughanem, and N. Aussenac-Gilles, “A Conceptual Indexing Approach based on Document Content Representation”, Dans: CoLIS5: Fifth International Conference on Conceptions of Libraries and Information Science, Glasgow, UK, 4 juin 8 juin 2005., F. Crestani, I. Ruthven (Eds.), Lecture Notes in Computer Science LNCS Volume 3507/2005, Springer-Verlag, Berlin Heidelberg, pp. 171-186.

[2] M. Baziz, M. Boughanem, and N. Aussenac-Gilles, “The Use of Ontology for Semantic Representation of Documents“, Dans: The 2nd Semantic Web and Information Retrieval Workshop(SWIR), SIGIR 2004, Sheffield UK, 29 juillet 2004, Ying Ding, Keith van Rijsbergen, Iad Ounis, Joemon Jose (Eds.), pp. 38-45.

[3] F. Boubekeur, M. Boughanem, and L. Tamine, “Exploiting association rules and ontology for semantic document indexing“, Dans: 12th International conference IPMU08, Information Processing and Management of Uncertainty in knowledge-Based Systems, Malaga, 22- 27, June 08, Spain, pp. 464-472.

[4] F. Boubekeur, M. Boughanem, and L.Tamine, “Semantic Information Retrieval Based on CP-Nets”, Dans: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007), London, 23/07/07- 26/07/07, IEEE, July 2007, pp. 1-7.

[5] M. Boughanem, I. Mallak, and H. Prade, “A new factor for computing the relevance of a document to a query”, Dans: IEEE World Congress on Computational Intelligence (WCCI 2010), Barcelone, July 2010, (to appear).

[6] M. Boughanem and C. Soulé-Dupuy, “A Connexionist Model for Information Retrieval“, DEXA 1992, pp. 260-265.

[7] P. Buitelaar and H. Uszkoreit, “MuchMore: Concept-Based Cross-Lingual Information Retrieval in the Medical Domain”, In: Kuenstliche Intelligenz, Heft 2/04, 2004, pp. 43-44, (http://muchmore.dfki.de/resources1.htm, 2010).

[8] M. Cuadros, JM., Atserias, J., M. Castillo, M., and G. Rigau, “Automatic acquisition of sense examples using exretriever“, In *IBERAMIA Workshop on Lexical Resources and The Web for Word Sense Disambiguation*. Puebla, Mexico, 2004, pp. 97-104.

[9] D. Dinh and L. Tamine, “Vers un modèle d'indexation sémantique adapté aux dossiers médicaux de patients“, Dans: Conférence francophone en Recherche d'Information et Applications (CORIA 2010), Sousse, Tunisia, March 2010, Hermès editions, (electronic support).

[10] E.A. Fox, “Extending the boolean and vector space models of information retrieval with p-norm queries and multiple concept types”, PhD thesis, Ithaca, NY, USA, 1983.

[11] J.A Guthrie, L. Guthrie, Y. Wilks, and H. Aidinejad, “Subject-dependant cooccurrence and word sense disambiguation“, In Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, Berkley, CA, pp. 146-152.

[12] B.Y. Kang and S.J. Lee, “Document indexing: a concept-based approach to term weight estimation“, In Journal of Information Processing & Management. Volume 41, Issue 5, September 2005, pp. 1065-1080.

[13] L.R. Khan, D. Mc Leod, and E.Hovy, “Retrieval effectiveness of an ontology-based model for information selection“, The VLDB Journal (2004)13, pp. 71–85.

[14] R. Krovetz, “Homonymy and polysemy in information retrieval”, In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97}, pp. 72-79.

[15] C. Leacock, G.A. Miller, and M. Chodorow, “Using corpus statistics and WordNet relations for sense identification“, Comput. Linguist. 24, 1 (March 1998), pp. 147-165.

[16] M.E. Lesk, “Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a nice cream cone”, In Proceedings of the SIGDOC Conference. Toronto, 1986, pp. 24-26.

[17] D. Lin, “An information-theoretic definition of similarity“, In Proceedings of 15th International Conference On Machine Learning, 1998, pp. 296-304.

[18] O. Medelyan, D. Milne, C. Legg and I.H. Witten, ”Mining meaning from Wikipedia“, In International Journal of Human-Computer Studies archive, Volume 67, Issue 9, September 2009, pp. 716-754, ISSN: 1071-5819.

[19] R. Mihalcea and D. Moldovan, “Semantic indexing using WordNet senses”, In Proceedings of ACL Workshop on IR & NLP, Hong Kong, October 2000, pp. 35-45.

[20] G. Miller, “WordNet: A Lexical database for English”, Actes de ACM 38, pp. 39-41.

[21] P. Resnik, “Disambiguating noun groupings with repect to WordNet senses”,, 3thWorkshop on Very Large Corpora, 1995, pp. 54–68.

[22] P. Resnik, “Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language”, Journal of Artificial Intelligence Research (JAIR), 11, 1999, pp. 95-130.

[23] S.E. Robertson, “The probability ranking principle in IR”, Journal of Documentation 33, 1977, pp. 294-304. Reprinted in: K. Sparck Jones and P. Willett (eds), Readings in Information Retrieval. Morgan Kaufmann, 1997, pp. 281-286.

[24] H. Schütze and J. Pedersen, “Information retrieval based on word senses”, In Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval, pp. 161-175.

[25] M. Sussna, “Word sense disambiguation for free-text indexing using a massive semantic network”, 2nd International Conference on Information and Knowledge Management (CIKM-1993), pp. 67–74.

[26] O. Uzuner, B. Katz, and D. Yuret, “Word Sense Disambiguation for Information Retrieval”, AAAI/IAAI 1999, pp. 985.

[27] J. Véronis and N. Ide., “Word sense disambiguation with very large neural networks extracted from machine readable dictionaries”, In 13th International Conference on Computational Linguistics (COLING-1990), 2, 1990, pp. 389-394.

[28] E. M. Voorhees, “Using WordNet to disambiguate word senses for text retrieval”, Association for Computing Machinery Special Interest Group on Information Retrieval. (ACM-SIGIR-1993): 16thAnnual International Conference on Research and Development in Information Retrieval, 1993, pp. 171-180.

[29] S.F. Weiss, “Learning to disambiguate”, In Information Storage and Retrieval, 9, 1973, pp. 33-41.

[30] Y. Wilks and M. Stevenson, “Combining independent knowledge source for word sense disambiguation”, Conference «Recent Advances in Natural Language Processing », 1997, pp. 1-7.

[31] D. Yarowsky., “Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora”, In Proceedings of the 14th International Conference on Computational Linguistics (COLING-92). Nantes, France, August 1992, pp. 454-460.

[32] D. Yarowsky, “Unsupervised word sense disambiguation rivaling supervised methods”, In 33rd Annual Meeting, Association for Computational Linguistics, Cambridge, Massachusetts, USA, 1995, pp. 189-196.

Semantic Workflow Adaption in Support of Workflow Diversity

Gregor Grambow and Roy Oberhauser
Computer Science Dept.
Aalen University
Aalen, Germany
{gregor.grambow, roy.oberhauser}@htw-aalen.de

Manfred Reichert
Institute for Databases and Information Systems
Ulm University
Ulm, Germany
manfred.reichert@uni-ulm.de

Abstract — **The application of business process execution and guidance to environments with highly dynamic situations and workflow diversity is hindered by rigid predefined workflow models. Software engineering environments constitute an acute example where developers could benefit from automated workflow guidance if the workflows were made sufficiently concrete and conformant to actual situations. A context-aware software engineering environment was developed utilizing semantic processing and situational method engineering to automatically adapt workflows utilizing an adaptive process-aware information system. Workflows are constructed via context knowledge congruent to the current situation. Preliminary results suggest this technique can be beneficial in addressing high workflow diversity while providing useable guidance and reducing workflow modeling effort.**

Keywords- application of semantic processing; domain-oriented semantic applications; automated workflow adaptation; situational method engineering; process-aware information systems; software engineering environments

I. INTRODUCTION

Business process management (BPM) and automated human process guidance have been shown to be beneficial in various industries [10][15]. However, BPMs often prerequisite and rigid model makes its application in highly dynamic and possibly evolving domains with diverse workflows, such as software engineering (SE), difficult. SE has multiform and divergent process models, unique projects, multifarious issues, a creative and intellectual process, and collaborative team interactions, all of which affect workflow models. These challenges have hitherto hindered automated concrete process guidance and often relegated processes to generalized and rather abstract process models (Open Unified Process, VM-XT, etc.) with inanimate documentation for guidance. Manual project-specific process model tailoring is typically done via documentation without investing in automated workflow guidance. Although automated workflows could assist overburdened software engineers by providing orientation and guidance for problems, guidance that does not coincide with the reality of the situation must be ignored and may cause the entire system to be mistrusted. Thus, adaption and pertinence to the dynamic and diverse SE situations is requisite for adoption of automated workflow guidance in SE environments (SEEs).

While classical application techniques may lend themselves to foreseeable common workflows with conformant sequences (*intrinsic workflows*), workflow integration for non-generalized diverse workflows that are external to the process model (*extrinsic*) presents a challenge. Considering SE, guidance is desirable for issues such as specialized refactoring, fixing bugs, etc., yet it is generally not feasible to pre-model workflows for SE issue processing, since SE issue types can vary greatly (tool problems, component versioning, merge problems, documentation inconsistencies, etc.). Either one complex workflow model with many branches is necessary that takes all cases into account, or many workflow variants need to be modeled, adapted, and maintained for such dynamic environments. The associated exorbitant expenditures thus limit workflow usage to well-known common sequences as typically seen with industrial BPM usage.

To briefly illustrate, SE issues that are not modeled in the standard process flow of defined SE processes (such as OpenUP [19] and VM-XT [25]) include bug fixing, refactoring, technology swapping, or infrastructural issues. Since there are so many different kinds of issues with ambiguous and subjective delineation, it is difficult and burdensome to universally and correctly model them in advance for acceptability and practicality. Many tasks may appear in multiple issues but are not necessarily required, bloating different SE issue workflows with many conditional tasks if pre-modeled. Figure 1 shows such a workflow just for bug fixing which is explained in the following.

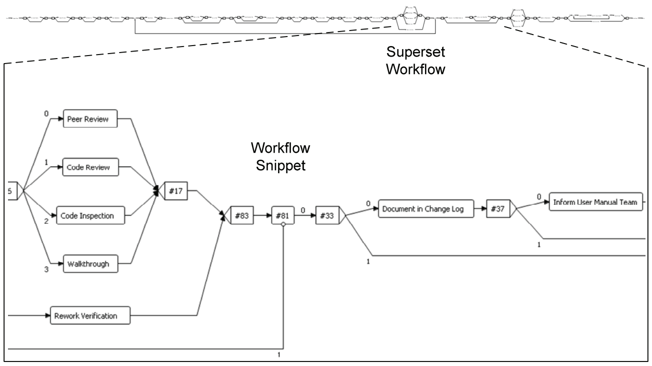


Figure 1. Example of pre-modeled workflow for bug fixing

The above workflow contains over 30 activities, and the snippet shows only different reviewing tasks from which one is chosen due to different project parameters (risk, urgency). Thereafter, it loops back if the post-review code or document rework was insufficient. The subsequent tasks deal with documenting the changes. Again, due to different project parameters the appropriate task has to be chosen, whereby none, one, or both of the tasks may be applicable.

The resulting workflow problems for environments such as SE are first that the exorbitant cost of modeling diverse workflows results in the absence of *extrinsic workflow* models and subsequently automated guidance for these types, yet these unique cases are often the ones where guidance is most desirable. Second, rigid, pre-defined workflow models are limited in their adaptability, thus the workflows become situationally irrelevant and are thus ignored. Third, entwining the complex modeling of situational property influences (as risk or urgency) on workflows within the workflows themselves incorporates an implicit modeling that unduly increases their complexity and makes correct maintenance difficult. The cognitive effort required to create and maintain large process models syntactically can lower the attention towards the incorporated semantic problem-oriented content.

Previous work has described a holistic approach that includes semantic technologies for SE lifecycles [18] and context-awareness [16], while this work focuses on applying context-awareness utilizing semantic processing and situational method engineering [22] for automatically adapting workflows in a process-aware information system. Support is provided for both *intrinsic workflows,* denoting workflows pre-modeled in archetype SE processes, and *extrinsic workflows,* indicating sets of activities not modeled in workflows of those archetype processes. The modeling of contextual property influences is transferred from the workflows themselves to an ontology, simplifying the modeling and making property effects explicit. Dynamic on-the-fly workflow generation and adaptation using contextual knowledge for a large set of diverse workflow variants is thus supported, enabling pertinent workflow guidance for workers in such environments.

The remainder of this paper is organized as follows: the solution approach is described in Section II. In Section III, the realization is portrayed and then evaluated in Section IV. Related work is discussed in Section V, followed by the conclusion.

II. Solution Approach

As a background to the solution approach, the incorporated frameworks that affected the environment and influenced the solution will first be discussed.

A. *Software Engineering Environment*

CoSEEEK (Context-aware Software Engineering Environment Event-driven frameworK) [16] consists of a hybrid semantic computing approach towards improved context-aware SEEs. The conceptual architecture is shown in Figure 2. *Event Extraction* consists of *SE Tool* sensor events (e.g., creation of a certain source code file) that are acquired and then stored in an *XML Tuple Space*, where it may optionally be annotated with relevant contextual information (e.g., link to a requirement for traceability). *Event Processing* detects higher-level events. This may result in workflow adjustment (e.g., according to the type of source code file, an activity Implement Solution or Implement Test may be chosen), and the software engineer is informed of a change in tasks via process management in their IDE (Integrated Development Environment). The *Context Module* includes an ontology and reasoner.

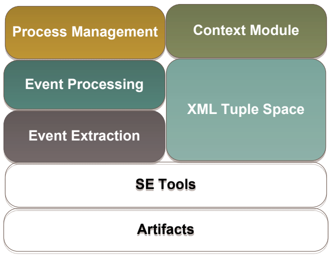


Figure 2. CoSEEEK conceptual architecture

Process Management requires an adaptable process-aware information system due to the dynamic nature of the problem the current approach seeks to address. Therefore the AristaFlow BPM suite (formerly ADEPT2) [3] was chosen for its realization. It allows authorized agents to dynamically adapt and evolve the structure of process models during runtime. Such dynamic process changes do not lead to an unstable system behavior, i.e., none of the guarantees achieved by formal checks at build-time are violated due to the dynamic change at runtime. Correctness is ensured in two stages. First, structural and behavioral soundness of the modified process model is guaranteed independent from whether or not the change is applied at the process instance level. Second, when performing structural schema changes at the process instance level, this must not lead to inconsistent or erroneous process states afterwards. AristaFlow applies well-elaborated correctness principles in this context [23]. Despite its comprehensive support for dynamic process changes, ADEPT2 has not considered automated workflow adaptations.

CoSEEEK provides comprehensive automated process support to address the aforementioned challenges. That implies workflows belonging to SE processes as well as workflows dealing with SE issues that are not modeled in those processes. While the automated support for *intrinsic workflows* is described in [17], the support for *extrinsic workflows* and the approach for their semantic problem-oriented modeling utilizing situational method engineering is the focus of this paper.

B. *Application of Situational Method Engineering*

Situational method engineering adapts generic methods to the actual situation of a project. This is done based on different influence factors called process properties which capture the impact of the current situation, and product properties which realize the impact of the product currently being processed (for this context the type of component, e.g., a GUI or database component). To strike a balance between rigidly pre-modeled workflows and no process guidance, the idea is to have a basic workflow for each case that is then dynamically extended with activities matching the current situation. The construction of the workflows utilizes a case base as well as a method repository. The case base contains a workflow skeleton for each different case. This workflow only contains the absolutely fundamental activities that are always executed for that case. The method repository contains all further activities whose execution is possible according to the case. To be able to choose the appropriate activities for the current artifact and situation, the activities are connected to properties that realize product and process properties of situational method engineering.

Each SE issue, such as refactoring or bug fixing, is mapped to exactly one case relating to exactly one workflow skeleton. To realize a pre-selection of activities (e.g., Create Branch or Code Review) which semantically match an issue, the issue is connected to the activity via an n-m relation. The activities are in turn connected to properties specifying the dependencies among them. The selection of an activity can depend on various process as well as product properties. To model the characteristic of an issue leading to the selection of concrete activities, the issue is connected to various properties. The properties have a computed value indicating the degree in which they apply to the current situation. An example for a property would be 'risk' with a value of 'very high', marking that issue as a very high-risk issue. Utilizing the connection of activity and property, selection rules for activities based on the values of the properties can be specified.

C. *Information Gathering*

To leverage the automatic support for *extrinsic workflows*, the computation of the values of the properties is a key factor. The approach presented in this paper unifies process and product properties in the concept of the property, which can be influenced by a wide range of factors. The integration of different modules and applications and the unification of various project areas in CoSEEEK enable automatic computation of the values comprising context knowledge. On the one hand, tool integration can provide meaningful information about the artifact that is processed in the current case. For example, if the artifact is a source code file, static code analysis tools such as PMD can be used to execute various measurements on that file, revealing various potential problems. If a high coupling factor was detected, this would raise the product property 'risk' associated to that file. On the other hand, the integration of various project areas like resource planning entails context knowledge about the entire development process. An example would be the raising of the property 'risk' if the person processing the current case is a junior engineer.

Both of these aspects deal with implicit information gathering. Since not all aspects of a case are necessarily covered by implicit information, and not all options for gaining knowledge about the case are always present, the system also utilizes explicit information gathering from the user processing the case. To enable and encourage the user to provide meaningful information, a simple response mechanism is integrated into the CoSEEEK GUI (to be shown in the next section). Via this mechanism, the user can directly influence process as well as product properties. To keep the number of adjustable parameters rather small, the concept of a product category was introduced. The product category unites the product properties in a pre-specified way. An example for this would be a database component versus a GUI component: the database component is likely to have more dependencies, whereas the GUI component presumably has more direct user impact. The influence of the product categories on the different properties is specified in advance and can be adapted to fit various projects. Selected process properties can be set directly. The computation of all other influences on the properties is explained in the following section.

D. *Activity selection and sequencing*

To be able to dynamically build up the workflow for an SE issue, after completing the computation of the property values activities have to be selected and placed in the correct order. This is done utilizing the connection between properties and activities. An activity can depend on one or more properties. Examples include selection rules such as:

- 'Choose activity code inspection if risk is very high and criticality is high and urgency is low' or
- 'Choose activity code review if risk is high and criticality is high'.

The selection of activities results in an unordered list of activities that have to be correctly sequenced and inserted into the workflow skeleton. To guarantee this, CoSEEEK uses a set of simple semantic constraints. These constraints do not only enforce which activities are permitted in a particular workflow, but also determine their correct ordering. A predefined sequencing of the activities is required since the workflow is built up at the beginning of its execution. That implies that each of the activities must have a binary relation to all other activities so that every possible set of activities can be sequenced. For the time being, the approach presented requires a proper specification of these constraints and only deals with linear workflows. Future work will concentrate on constraints that are more complex and the integration of workflow patterns. Table I enumerates the constraints currently used.

TABLE I. ACTIVITY SEQUENCING CONSTRAINTS

Constraint	Meaning
X before Y	if X and Y are present, X should appear before Y
X after Y	if X and Y are present, X should appear after Y
required after	if Y is present X must also be present, after Y
required before	if Y is present X must also be present, before Y
mutual exclusion	if X is present the presence of Y is prohibited

Structural integrity of the workflows is guaranteed based on the built-in mechanisms of AristaFlow, which imply correctness checks for each change operation applied to the workflow as discussed in [3].

E. Concrete Procedure

The concrete procedure for the handling of an SE issue in the presented approach is as follows. As entry point for the workflow there is an event in the framework indicating that an SE issue is assigned to a user. This event can come from various sources. Examples include the assignment of an SE issue to a person in a bug tracker system or the manual triggering by a user via the GUI. The next step is the determination of a case for that issue like ‘Bug fixing’ or ‘Refactoring’. Depending on the origin of the event, this can be done implicitly or explicitly by the user.

When the case is specified, the workflow starts for the user using the workflow skeleton assigned to that case, as does the contextual information-gathering phase for the properties of the case.

After having determined the properties for the case, the additional activities matching the current situation and product are selected. This set of activities is then checked for integrity and correctly sequenced utilizing semantic constraints. Subsequently, the activities are integrated into the running process instance.

If one or more of the properties change during the execution of the workflow, the prospective activities are deleted (if still possible) and a new sequence of activities is computed for the rest of the workflow.

III. TECHNICAL REALIZATION

This section describes the concrete implementation of the SE issue process explained in the preceding section.

All communication between the modules is performed using an XML implementation of the Tuple Space paradigm [6] on top of the eXist XML database [5] for event storage and Apache CXF for web service communication. To enable CoSEEEK to receive events from external SE tools, the Hackystat framework [8] is used, which provides a rich set of sensors for various applications. In the concrete case, the bug tracker Mantis is used in conjunction with a sensor that generates an event when an SE issue is assigned to a person. That event contains information about the kind of issue for case selection and about the person. In case of a real ad hoc issue that is not recorded in a bug tracker, the event for instantiation of an issue workflow can be triggered from the GUI as well, requiring the user to select a case manually.

The event is then automatically received by the process module which instantiates a skeleton workflow based on the process template relating to the selected case. The activity components of AristaFlow for these workflows are customized to communicate over the Tuple Space and thus enable user interaction during the execution of each task. The first task of each SE issue is ‘Analyze Issue’ to let the user gain knowledge about the issue and provide information about process and product properties to the system via the GUI. The GUI is a lightweight web interface developed in PHP that can be executed in a web browser as well as preferably directly in the users IDE. Figure 3 shows the GUI enabling the user to directly set process properties and to choose a product category that affects product properties. On the lower part of the GUI, the current task is shown as well as one possible upcoming task from other workflows the user is working in. In that way, task switching is facilitated without subjecting the user to information overload showing all available tasks of all open workflows. Via the dropdown list at the bottom of the GUI, the user can switch between available tasks for the case when the pre-selection is inappropriate.

The Context module has three main responsibilities: it realizes the case base, the method repository, and contains context information about the entire project. This information is stored in an OWL-DL [28] ontology to unify the project knowledge and enable reasoning over it. The use of an ontology reduces portability, flexibility and information sharing problems that are often coupled to relational databases. Additionally, ontologies facilitate extensibility since they are, in contrast to relational databases, based on an open world assumption and thus allow the modeling of incomplete knowledge. To programmatically access the ontology, the Jena API [13] is used within the Context Module.

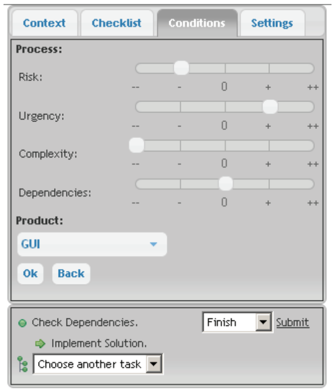


Figure 3. GUI for property acquisition

The adaptation of the running instances works as follows: The skeleton process is instantiated, offering the user the aforementioned ‘Analyze Issue’ task to provide information. The information from the user is encapsulated in an event that is received by the process module. The process module queries the context module, which provides the set of activities to be inserted in the process instance and performs the adaptation. Thus, the process is already aligned to the current situation and product when the user continues.

A. Context Module

This subsection describes how the context module utilizes the ontology to derive property values and select appropriate activities. To leverage real contextual-awareness, the ontology features various concepts for different areas of a project. These are semantic enhancements to process management utilized for *intrinsic workflows*, quality management, project staffing, or traceability. For process management the concepts of *Activity*, *Workflow*, *Assignment,* and *AtomicTask* are used to enrich processes, activities, and tasks with semantic information. Quality management features the concepts of the *Metric*, *Measure*, *Problem*, *Risk*, *Severity* and *KPI* (key performance indicator) to incorporate and manage quality aspects in the project context. The concepts of *Person*, *Team*, *Role*, *Effort*, *SkillLevel* and *Tool* are integrated to connect project staffing with other parts of the project. To further integrate all project areas and facilitate a comprehensive end-to-end traceability the concepts of *Tag* and *Event* can be connected and used in conjunction with all other concepts. Due to space limitations, only the concepts directly relevant to the discussion of *extrinsic workflows* are explained. Figure 4 illustrates the relating classes in the ontology.

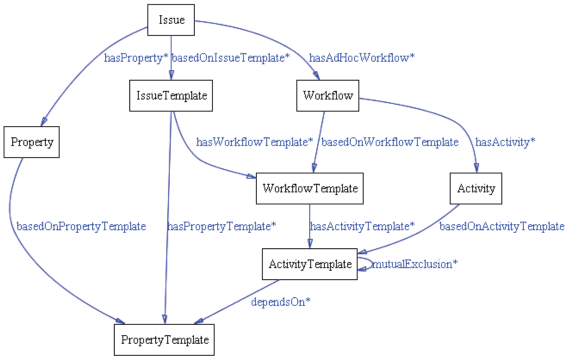


Figure 4. Classes in the Ontology

To predefine the different SE issues, a set of template classes has been defined with their skeleton workflows and activities as well as the properties applying to them. Each *IssueTemplate* is connected to a *WorkflowTemplate* that stores the information about the concrete process template in AristaFlow and is in turn connected to multiple *ActivityTemplates*. These define the set of possible *Activities* that can be inserted in the *Workflow* of that issue. The *IssueTemplate* is also connected to one or more *PropertyTemplates*, yielding the capability to specify not only a unique set of *Activities* for each *Issue*, but also a unique set of *Properties* with a unique relation to the *Activities*.

When a new SE *Issue* is instantiated, it derives the *Workflow* and the *Properties* from its associated *IssueTemplate*. Each *Property* holds a value indicating how much this *Property* applies to the current situation. These values can be influenced by various factors that are also defined by the *PropertyTemplate*. Figure 5 exemplifies three different kinds of influences that are currently used. Future work will include the integration of further concepts of the ontology influencing the *Properties* as well as extending the ontology to further leverage the context knowledge available to CoSEEEK.

The *ProductCategory* specified in the GUI has a direct influence on the product *Properties*. Furthermore, there can be *Problems* relating to the processed *Artifact* indicated by violations of metrics. The *SkillLevel* of the *Person* dealing with the SE *Issue* serves as example for an influence on the process properties here. There are four possible relations between entities affecting the *Properties* and the *Properties* capturing strong and weak negative as well as positive impacts (where Figure 5 only shows the weak ones, 'enhances' and 'deteriorates'). These are all used to compute the values of the *Properties*. The values are initialized with '0 (neutral)' and incremented / decremented by one or two based on the relations to the different influences. The values are limited to a range from '-2 (very low)' to '2 (very high)', thus representing five possible states for the degree to which the property applies to the current situation.

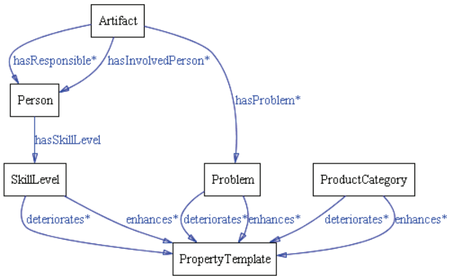


Figure 5. Influences on Properties

To select appropriate *Activities* according to the current properties, six possible connections are utilized. These are 'weaklyDependsOn', 'stronglyDependsOn' and 'dependsOn', meaning the *Activity* is suitable if the value of the *Property* is '1 (high)' or '2 (very high)', or just positive and the other three connections for negative values (for simplicity, Figure 4 only shows 'dependsOn'). Each *Activity* can be connected to multiple *Properties*. Based on an *Issue*, for each attributed *ActivityTemplate* a SPARQL query is dynamically generated which returns the corresponding Activity if the *Properties* of the current situation match. Listing 1 shows such a query for an *Activity* 'act' that is based on an *ActivityTemplate* 'at' and depends on two different *Properties* 'prop1' and 'prop2' which are, in turn, based on *PropertyTemplates* 'pt1' and 'pt2'.

Listing 1 Activity selection SPARQL query

```
PREFIX project:
<http://www.htw-aalen.de/coseeek/context.owl#>
SELECT ?act
WHERE {
   ?act project:basedOnActivityTemplate ?at.
   ?at project:title "AT_CodeReview".
   ?issue project:title "CodeFixRequired".
   ?issue project:hasProperty ?prop.
   ?prop project:basedOnPropertyTemplate ?pt.
   ?at project:weaklyDependsOn ?pt.
   ?prop project:weight "1".
   ?issue project:hasProperty ?prop2.
   ?prop2 project:basedOnPropertyTemplate ?pt2.
   ?at project:stronglyDependsOn ?pt2.
   ?prop2 project:weight "2".}
```

Lastly, the semantic constraints are mapped to connections between *ActivityTemplates* (for simplicity, Figure 4 only shows 'mutualExclusion'). To guarantee semantic correctness, the algorithm first checks if all required activities are in place or if a mutual exclusion constraint is violated. Utilizing the before / after constraints, the sequencing is finally done via a simple sorting of the list of activities. For simplicity, an abstraction from workflow patterns such as loops or decisions is made here.

The significance of this contribution is on the one hand that SE issue workflows that are extrinsic to archetype SE processes are not only explicitly modeled, but also dynamically adapted to the current issue and situation based on various properties derived from the current product, the context, and the user. Thus, it is possible to provide situational and tailored support and guidance for software engineers processing SE issues. On the other hand, the proposed approach shows promise for improvement and simplification of process definition activities for *extrinsic workflows*. The initial effort to define all the activities, issues, properties, and skeleton workflows may not be less than predefining huge workflows for the issues, but the reuse of the different concepts is furthered. Thereafter the creation of new issues is simplified since they only need to be connected to activities they should contain that are later automatically inserted to match the current situation. Yet the main advantage is of a semantic nature: the process of issue creation is much more problem-oriented using the concepts in the ontology versus creating immense process models. The process engineer can concentrate on activities matching the properties of different situations rather than investing cognitive effort in the creation of huge rigid process models matching every possible situation. Likewise, the analysis of issues allows simple queries to the ontology returning problem-oriented knowledge such as 'Which activities apply to which issues' or 'Which activities are applied for high risk time critical situations'.

IV. Evaluation

This section illustrates the advantages of the proposed approach via a synthetic but concrete practical scenario generated in a lab environment to ascertain scalability and performance of the initial approach using different measurements. It remains difficult to prove the applicability of the approach for the majority of real world SE use cases, thus future work will include practical case studies utilizing CoSEEEK with industrial partners of the research project. Additionally, CoSEEEK is in use by the CoSEEEK development team itself for the development of CoSEEEK.

A. Scenario Solved

The concrete scenario considered shows two possible generated workflows for the bug fix issue presented in Section I. For this scenario, a set of properties has been defined as well as activities and their dependencies on the properties. The first case deals with a fix of a GUI component. That component is assumed to be part of a simple screen not often used by customers. The second case deals with a database component. The fix is assumed to have an impact on multiple tables in the database. Table II depicts the chosen properties for the cases as well as the values that were chosen for them by the developer via the CoSEEEK web GUI. It is assumed that no other influences exist for the properties.The chosen values lead to the selection of different activities for the different workflows. For instance, due to the direct user impact of the GUI component, the activity Document in Patch/Release Change Log has been chosen as illustrated in Figure 6 (the generated workflow has been rearranged for better readability).

TABLE II. EXAMPLE SME PROPERTIES OF CASES

	Component	GUI (Case 1)	DB (Case 2)
Product Properties	criticality	o	+
	user impact	++	o
	dependencies	-	+
	complexity	o	+
	risk	o	+
Process Properties	risk	-	o
	urgency	o	-
	complexity	-	+
	dependencies	o	o

Due to the risk and complexity of the database component and the task relating to it, the creation of a separate branch as well as a code review and the explicit check for dependencies have been prescribed as depicted Figure 7.

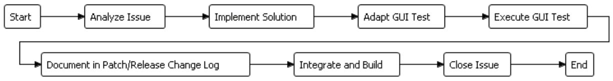


Figure 6. Example Workflow GUI Component

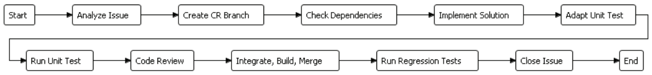


Figure 7. Example Workflow Database Component

Ignoring the abstraction from workflow patterns, they are nevertheless much simpler than the pre-modeled example mentioned in the Problem Scenario section. This automated adaption thus supports workflow diversity, reducing complexity and maintenance compared to all-encompassing models. The scenario illustrates the usefulness of the guidance via the chosen activities by these two considerable different workflows containing tasks matching the situation as well as the processed artifact. Future case studies will be used to further evaluate the usefulness of the workflows and to refine the properties and their relation to the activities.

B. Performance Measurement

Due to space limitations, only the area of the concept that is likely to have the greatest performance impact was selected for measurement. This is the sequencing of the concrete activities based on the constraints in the context module. For performance testing, the test system consisted of an AMD dual core Opteron 2.4 GHz processor, 3.2GB RAM, Windows XP Pro SP3, and Java Runtime Environment 1.5.0_20. All measurements were executed five times consecutively using the average of the last three measurements.

The sequencing of the activities is separated into two parts to yield better runtime performance: when a new SE issue is defined via the issue template or the number of the attributed activity templates changes, an indexing procedure is started. This procedure uses the 'after', 'before', 'requiredAfter', and 'requiredBefore' constraints to generate a simple index for all activity templates for one issue template. The index is later used for the concretely selected activities of an issue to accelerate the sequencing. For the measurement, it is assumed that half of the activities that are possible for one issue have been selected. Table III depicts the measured total values for indexing and sequencing for different numbers of activities. The values show that after the indexing, which happens usually only once after the definition of an issue type, the sequencing is not resource-intensive.

TABLE III. CONTEXT MODULE LATENCY MEASUREMENTS

Number of activity templates per issue	Indexing latency (ms)	Number of activities per issue	Sequencing latency (ms)
10	5	5	0
50	15	25	0
100	41	50	7
500	890	250	11
1000	3532	500	15

As can be seen from the plain increase of computation times, the results show adequate performance for the CoSEEEK approach.

V. RELATED WORK

The combination of semantic technology and process management technology has been used in various approaches. The concept described in [9] utilizes the combination of Petri Nets and an ontology to achieve machine readable process models for better integration and automation. This is achieved creating direct mappings of Petri Net concepts in the ontology. The main focus of the approach presented in [11] is the facilitation of process models across various model representations and languages. It features multiple levels of semantic annotations as the meta-model annotation, the model content annotation, and the model profile annotation as well as a process template modeling language. The approach described in [12] presents a semantic business process repository to automate the business process lifecycle. Its features include checking in and out as well as locking capabilities and options for simple querying and reasoning that is more complex. Business process analysis is the main focus of COBRA presented in [21]. It develops a core ontology for business process analysis with the aim to provide better easier analysis of processes to comply with standards or laws like the Sarbanes-Oxley act. The approach described in [26] proposes the combination of semantic and agent technology to monitor business processes, yielding an effective method for managing and evaluating business processes. These approaches feature a process-management-centric use of semantic technology, while CoSEEEK not only aims to further integrate process management with semantic technology; it also integrates contextual information on a semantic level producing novel synergies alongside new opportunities for problem-oriented process management.

With regard to automatic workflow support and coordination, several approaches exist. CASDE [7] utilizes activity theory to provide a role-based awareness module managing mutual awareness of different roles in the project. CAISE [2], a collaborative SE framework, enables the integration of SE tools and the development of new SE tools based on collaboration patterns. Caramba [4] features support for ad hoc workflows utilizing connections between different artifacts, resources, and processes to provide coordination of virtual teams. UML activity diagram notation is used for pre-modeled workflows. For ad hoc workflows not matching a template, an empty process is instantiated. In that case, work between different project members is coordinated via so-called Organizational Objects. These approaches primarily focus on the coordination of dependencies between different project members and do not provide unified, context-aware process guidance incorporating *intrinsic* as well as *extrinsic workflows*.

The problem of rigid processes unaligned to the actual situation is addressed in different ways by approaches like Worklets [1], DECLARE [20], Agentwork [13], or Pockets of Flexibility (PoF) [24]. Worklets feature the capability of binding sub-process fragments or services to activities at runtime, thus not enforcing concrete binding at design time. DECLARE provides a constraint-based model that enables any sequencing of activities at runtime as long as no constraint is violated. A combination of predefined process models and constraint-based declarative modeling has been proposed in [24], wherein at certain points in the defined process model (called Pockets of Flexibility) it is not exactly defined at design time which activities should be executed in which sequence. For such a PoF, a set of possible activities and a set of constraints are defined enabling some runtime flexibility. However, the focus of DECLARE as well as PoF is on the constraint-based composition and execution of workflows by end users, and less on automatic workflow adaptations. Agentwork features automatic process adaptations utilizing predefined but flexible process models, building upon ADEPT1 technology. The adaptations are realized via agent technology and used to cope with exceptions in the process at runtime. As opposed to the CoSEEEK approach, these approaches do not utilize semantic processing and do not incorporate a holistic project-context unifying knowledge from various project areas. For a complete discussion of flexibility issues in the process lifecycle, we refer to [27].

VI. CONCLUSION

The SE domain epitomizes the challenge that automated adaptive workflow systems face. Since SE is a relatively young discipline, automated process enactment in real projects is often not mature. One of the issues herein is the gap between the top-down abstract archetype SE process models that lack automated support and guidance for real enactment, and exactly the actual execution with its bottom-up nature. An important factor affecting this problem are activities belonging to specialized issues such as bug fixing or refactoring. These are on the one hand not covered by archetype SE processes and are on the other hand often so variegated that pre-modeling them is not feasible or currently cost-effective.

The synergistic CoSEEEK approach automatically adapts workflows in a process-aware information system by combining semantic-based SEE context knowledge with situational method engineering and automated process instance adaptations. SE issue processing is decomposed into various activities influenced by different process and product properties dependent on the actual situation, the project context knowledge, and the product that is the subject of the current SE issue. Based on these properties, an issue workflow is constructed automatically, dynamically, and uniquely for every SE issue. By combining a case base with a method repository, all activities that are requisite for an issue are automatically included, avoiding the necessity of building the current workflow from scratch.

The broader application of this approach would benefit domains similar to SE that exhibit dynamics and high workflow diversity with adaptable workflows for uncommon workflows, providing useable context-relevant guidance while reducing workflow modeling effort and maintenance by modeling influences outside of the workflows themselves.

Future work will consider issue learning for automated tailoring of process templates and to reduce external user information needs, continuous adaptation of product properties, automated case-learning, and process analysis of executed workflow instances.

ACKNOWLEDGMENTS

The authors wish to thank Stefan Lorenz for his assistance with implementation details. This work was sponsored by the BMBF (Federal Ministry of Education and Research) of the Federal Republic of Germany under Contract No. 17N4809.

REFERENCES

[1] Adams, M., ter Hofstede, A., Edmond, D., and van der Aalst, W., 'Worklets: A service-oriented implementation of dynamic flexibility in workflows,' In: Proc. Coopis'06, LNCS 4275, pp 291-308 (2006)

[2] Cook, C., Churcher, N., and Irwin, W., 'Towards Synchronous Collaborative Software Engineering,' in Proceedings of the Eleventh Asia-Pacific Software Engineering Conference. Busan, Korea, pp. 230-239 (2004)

[3] Dadam, P. and Reichert, M., 'The ADEPT Project: A Decade of Research and Development for Robust and Flexible Process Support - Challenges and Achievements,' Springer, Computer Science - Research and Development, 23(2), pp. 81-97 (2009)

[4] Dustdar, S., 'Caramba—A Process-Aware Collaboration System Supporting Ad hoc and Collaborative Processes in Virtual Teams,' in Distributed and Parallel Databases Vol. 15, Issue 1, Kluwer Academic Publishers Hingham, MA, USA (2004)

[5] Meier, W., 'eXist: An Open Source Native XML Database,' in Web, Web-Services, and Database Systems, Springer, LNCS vol. 2593/2009, pp. 169-183 (2009)

[6] Gelernter, D., 'Generative communication in Linda,' ACM Transactions on Programming Languages and Systems, 7(1):80-112, January 1985 (1985)

[7] Jiang, T., Ying, J., and Wu, M., 'CASDE: An Environment for Collaborative Software Development,' in Computer Supported Cooperative Work in Design III, Springer, pp. 367-376 (2007)

[8] Johnson, P.M., 'Requirement and Design Trade-offs in Hackystat: An In-Process Software Engineering Measurement and Analysis System,' in Proc. of the 1st Int. Symp. on Empirical Software Engineering and Measurement, IEEE Comp. Soc., pp. 81-90 (2007)

[9] Koschmider , A. and Oberweis, A., 'Ontology based Business Process Description,' in Proceedings of the CAiSE´05 workshops, 2005

[10] Lenz, R. and Reichert, M., 'IT Support for Healthcare Processes - Premises, Challenges, Perspectives,' Data and Knowledge Engineering, 61(1): pp. 39-58 (2007)

[11] Lin, Y. and Strasunskas, D., 'Ontology-based Semantic Annotation of Process Templates for Reuse,' in Proceedings of the 10th International Workshop on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD'05), 2005

[12] Ma, Z., Wetzstein, B., Anicic, D., and Heymans, S., and Leymann, F.,. 'Semantic Business Process Repository' In *Proc. of the Workshop on Semantic Business Process and Product Lifecycle Management*, 2007

[13] McBride, B., 'Jena: a semantic web toolkit,' Internet Computing, Dec. 2002

[14] Müller, R., Greiner, U., and Rahm, E., 'AGENTWORK: A Workflow-System Supporting Rule-Based Workflow Adaptation,' Data Knowl. Eng. 51(2), pp. 223-256 (2004)

[15] Müller, D., Herbst, J., Hammori, M., and Reichert, M., 'IT Support for Release Management Processes in the Automotive Industry,' Proc. 4th Int'l Conf. on Business Process Management (BPM'06), Vienna, LNCS 4102, pp. 368-377 (2006)

[16] Oberhauser, R., 'Leveraging Semantic Web Computing for Context-Aware Software Engineering Environments,' In G. Wu (ed.) Semantic Web, In-Tech, Vienna, Austria, 2010, pp. 157-179 (2010)

[17] Oberhauser, R., 'Towards Automated Test Practice Detection and Governance,' Int. Conf. on Advances in System Testing and Validation Lifecycle, pp. 19-24 (2009)

[18] Oberhauser, R. and Schmidt, R., 'Towards a Holistic Integration of Software Lifecycle Processes using the Semantic Web,' In: Proc. 2nd Int. Conf. on Software and Data Technologies (ICSOFT'07), Vol. 3, 2007, pp. 137-144 (2007)

[19] OpenUP, http://epf.eclipse.org/wikis/openup/ [June 2010]

[20] Pesic, M., Schonenberg, H., and van der Aalst, W.M.P., 'DECLARE: Full Support for Loosely-Structured Processes,' Proc. 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), pp. 287-298. Annapolis (2007)

[21] Pedrinaci, C., Domingue, J., and Alves de Medeiros, A.: 'A Core Ontology for Business Process Analysis' (2008), pp. 49-64

[22] Ralyté; J., Brinkkemper, S. and Henderson-Sellers, B. (Eds.), 'Situational Method Engineering: Fundamentals and Experiences,' Proc. IFIP WG 8.1 Working Conference, Sep. 2007, Geneva (2007)

[23] Rinderle-Ma, S., Reichert, M., and Weber, B., 'Relaxed Compliance Notions in Adaptive Process Management Systems,' In: Proc. 27th Int'l Conf. on Conceptual Modeling (ER'08), October 2008, Barcelona, LNCS 5231, pp. 232-247 (2008)

[24] Sadiq, S., Sadiq, W., and Orlowska, M., 'A framework for constraint specification and validation in flexible workflows,' Information Systems 30(5):349 – 378 (2005)

[25] Rausch, A., Bartelt, C., Ternité, T., and Kuhrmann, M., 'The V-Modell XT Applied - Model-Driven and Document-Centric Development,' in 3rd World Congress for Software Quality, Vol. III, Online Supplement, pp. 131—138 (2005)

[26] Thomas, M., Redmond, R., Yoon, V., and Singh, R., 'A Semantic Approach to Monitor Business Process Performance,' Communications of the ACM, pp. 55-59, 2005

[27] Weber, B.; Sadiq, S., and Reichert, M., 'Beyond Rigidity - Dynamic Process Lifecycle Support: A Survey on Dynamic Changes in Process-aware Information Systems,' Computer Science - Research and Development, 23(2): 47-65 (2009)

[28] World Wide Web Consortium, 'OWL Web Ontology Language Semantics and Abstract Syntax,' (2004) [June 2010]

Semantic-based Geographical Matchmaking in Ubiquitous Computing

Michele Ruta, Floriano Scioscia, Eugenio Di Sciascio, Giacomo Piscitelli
Politecnico di Bari
Via Re David 200
I-70125, Bari, ITALY
{m.ruta, f.scioscia, disciascio, piscitel}@poliba.it

***Abstract*—A full exploitation of semantics in mobile environments enhances discovery effectiveness allowing the instant fruition of services and resources. Hence, nowadays ever increasing efforts are spent in making available tools able to exploit Semantic Web techniques and technologies also in ubiquitous computing. This paper presents a platform-independent mobile semantic discovery framework as well as a working prototypical implementation, which enables advanced knowledge-based services taking into account user's location. The proposed approach is explained and motivated in a ubiquitous tourism case study, where some early evaluations are presented to prove its feasibility and usefulness.**

Keywords-Semantic Web, Ubiquitous Computing, Location-based Services, Resource Discovery.

I. INTRODUCTION

Techniques and ideas of the Semantic Web initiative are potential means to give flexibility to discovery [1]. In fact, Semantic Web technologies applied to resource retrieval open new possibilities, including: (i) formalization of annotated descriptions that become machine understandable so enabling interoperability; (ii) reasoning on descriptions and inference of new knowledge; (iii) validity of the Open World Assumption (OWA) (what is not specified has not to be interpreted as a constraint of absence) [2], overcoming limits of structured data models.

Though interesting results have been obtained in the evolution of canonical service discovery in the Web, several issues are still present in ad-hoc and ubiquitous environments, because of both host mobility and limited capabilities of mobile devices. Hence, many people equipped with handheld devices usually prefer traditional fixed discovery channels so renouncing to an instant fruition of resources or services. Nevertheless, the rising potentialities of wireless-enabled handheld devices today open new possibilities for implementing flexible discovery frameworks.

This paper presents a general approach for a semantic-based discovery in ubiquitous environments. It has been implemented in a Decision Support System (DSS), presented here with reference to a u-tourism (ubiquitous tourism) [3] case study. Users equipped with handheld devices can exploit semantic resource annotation to obtain a logic-based ranking and explanation of results, while all technicalities are hidden from them. Furthermore, a selective discovery is performed based on proximity measures. In the proposed touristic virtual guide application, this feature has been implemented by integrating a positioning module in the discovery tool. The application recognizes the user location and grades matchmaking outcomes according to geographical criteria, presenting an intuitive Graphical User Interface (GUI).

Main features of the proposed approach are: (i) semantic-based ranking of retrieved resources; (ii) full exploitation of non-standard inferences presented in [4] to enable refinement of user requests; (iii) fully graphical interface usable with no prior knowledge of logic principles; (iv) no physical space-temporal constraints in system exploitation. The interest domain is modeled with an OWL (Web Ontology Language) [5] ontology.

The remaining of the paper is structured as follows: in the next section, relevant background is revised; Section 3 describes framework and approach with the aid of the case study in Section 4. Some evaluations about the system are reported in Section 5 before concluding the paper.

II. BACKGROUND

The reader is supposed to be familiar with Description Logics (DLs), a family of logic formalisms for Knowledge Representation [2]. In this paper we will refer to the $\mathcal{ALN}$ (Attributive Language with Unqualified Number Restrictions) Description Logic, a subset of OWL DL having polynomial computational complexity for standard and non-standard inferences. Hereafter, for the sake of brevity, examples will be formalized by adopting DL syntax, whereas in our prototypes we exploit DIG (a syntactic variant of OWL) [6] because it is less verbose than OWL.

DL reasoners provide at least two basic standard inference services: concept *subsumption* (a.k.a. classification) and concept *satisfiability* (a.k.a. consistency) [2]. Given *R* (for Request) and *O* (for Offered resource), both consistent w.r.t. a common ontology $\mathcal{T}$ (containing axioms modeling domain knowledge), logic-based approaches to matchmaking in literature [7] use classification and consistency to grade match results in five categories: (i) *exact* - every feature requested in *R* is exactly the same provided by *O* and vice versa; (ii) *full-subsumption* - every feature requested in *R* is contained in *O*; (iii) *plug-in* - every feature offered in *O* is contained in *R;* (iv) *potential-intersection* - there is an intersection and no conflicts between the features offered in *O* and the ones requested in *R*; (v) *partial-disjoint* - some features requested in R are conflicting with some offered in *O*.

ISBN: 978-1-61208-104-5

Exact and full matches are the best ones for requesters, but they are infrequent in practical scenarios. A sequence of potential and partial matches is more likely. In [8], *Concept Abduction Problem (CAP)* and *Concept Contraction Problem (CCP)* were introduced and defined as non standard inferences for DLs. They allow to compute a logic-based ranking of potential and partial matches best approximating the request. Furthermore, they provide explanation of matchmaking outcomes, which is highly desirable to justify results so increasing user confidence in the DSS.

A noticeable feature is the exploitation of the above inference services w.r.t. an Open World semantics. If *R* and *O* are in potential match, the characteristics *B* (for *bonus*) [9] specified in *O* but not requested in *R* represent knowledge that can be elicited and proposed to the requester in order to refine her initial query. *B* can be computed by solving a CAP [9]. The bonus characteristics represent information the user might not be aware of or she initially considers not relevant. Hence, they are very useful in a query refinement process.

A. Related Work

Significant research and industry efforts have been focusing on service/resource discovery in mobile and ubiquitous computing. The main challenge is to provide paradigms and techniques that are effective and flexible, yet intuitive enough to be of practical interest for a potentially wide user base.

In [10], a prototypical mobile client is presented for semantic-based mobile service discovery. An adaptive graph-based representation allows OWL ontology browsing. However, a large screen seems to be required to explore ontologies of moderate complexity with reasonable comfort. Also preference specification requires a rather long interaction process, which could be impractical in mobile scenarios. Authors acknowledged these issues and introduced heuristic mechanisms to simplify interaction, *e.g.*, the adoption of default values.

In [11] a location- and context-aware mobile Semantic Web client is proposed for tourism scenarios. The goal of integrating multiple information domains has led to a division of the user interface into many small sections, whose clarity and practical usability seem questionable. Moreover, knowledge is extracted from several independent sources to build a centralized RDF triple store accessible through the Internet. The proposed architecture is therefore hardly adaptable to mobile ad-hoc environments.

Peer-to-peer interaction paradigms are actually necessary for fully decentralized semantic-based discovery infrastructures. Hence, mobile hosts themselves should be endowed with reasoning capabilities. Pocket KRHyper [12] was the first available reasoning engine for mobile devices. It provides satisfiability and subsumption inference services, which have been exploited by authors in a DL-based matchmaking between user profiles and descriptions of resources/services [13]. A limitation of that prototype is that it does not allow explicit explanation of outcomes. More recently, in [14] an embedded DL reasoning engine was presented in a mobile dating application, though applicable to other discovery scenarios. It acts as a mobile semantic matchmaker, exploiting non-standard inference services also used in the present framework. Semantically annotated personal profiles are exchanged via Bluetooth and matched with preferences of mobile phone users, to discover suitable partners in the neighborhood.

Due to the resource constraints of mobile devices, as well as to the choice of a cross-platform runtime environment, both the above solutions privilege simplicity of managed resource/service descriptions over expressiveness and flexibility. We conjecture that a native language optimized implementation can provide acceptable performance for larger ontologies and more resource-intensive inferences.

III. System Outline

Figure 1 shows the system architecture. A classical client/server paradigm is adopted: in our current prototype the resource provider is a fixed host over the Internet, exposing an enhanced DIG interface; the mobile client is connected through wireless technologies, such as IEEE 802.11 or UMTS/CDMA.

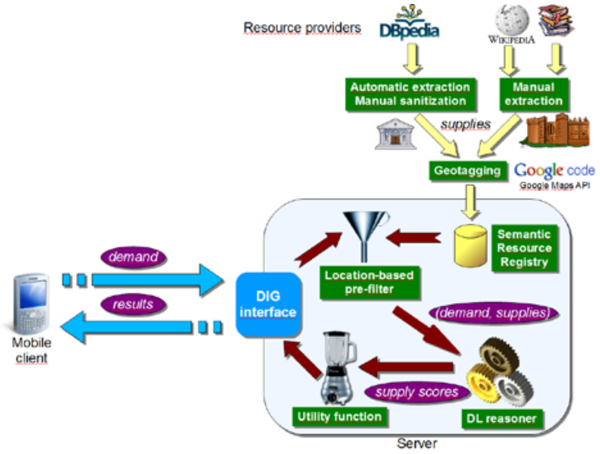


Figure 1. Architecture of the system prototype.

Available resources (*supplies*) were collected from several sources. The DBpedia RDF Knowledge Base (available at http://wiki.dbpedia.org), which is an extract of structured information from Wikipedia, was used to automatically obtain relevant information for many entries. DBpedia is a prominent example of the Linked Data effort [15], aimed at publishing structured data on the Web and to connect data between different data sources. RDF provides the semantic framework that allows both data to be machine understandable and related concepts from different datasets to be related to each other. Tens of datasets are already available, collectively containing several billion RDF statements and covering diverse application domains such as: encyclopedic, artistic and literary topics; healthcare, environmental and governmental data and statistics; commerce and finance. Resource providers can build innovative solutions, like the one presented here, upon these public Knowledge Bases (KBs).

RDF documents concerning resources of interest were directly retrieved from the KB using SPARQL query language. Obtained profiles were then sanitized (e.g., by removing textual abstracts, redundant and unnecessary information) and aligned to custom ontology for the cultural heritage domain through a semi-automatic procedure. Then each semantic annotation was geographically tagged exploiting the Google Maps API. Finally, all resources were stored into a *semantic registry* whose records contain: (i) a semantic annotation (in DIG language); (ii) a numeric ontology identifier, marking the domain ontology the annotation refers to; (iii) a set of data-oriented attributes manageable by proper *utility functions* (see later on for further details) and depending on the specific application; particularly, the tool proposed here requires a *(latitude, longitude)* pair of geographical coordinates; (iv) a set of user-oriented attributes. In the current prototype, each resource is supplied with a picture and a textual description.

On the client side, the user focuses on a given scenario early selecting the reference terminology. So she determines a specific context for the following interactions with the system. Different sessions in the application exploitation could refer to different ontologies and then could entail interactions aimed at different purposes. For example, a generic user could exploit the application as a pocket virtual guide for tourist purposes selecting a cultural heritage ontology and in a further phase, after concluding her visit, she can adopt it as a mobile shopping assistant to buy goods in a B2C (Business to Consumer) mobile marketplace: in that case she will select an e-commerce ontology.

Matchmaking can be carried out only among requests and supplied resources sharing the semantics of descriptions, *i.e.*, referring to the same ontology. Hence a preliminary agreement between client and server is required. Ontology identifiers are used for this purpose [16]. Then the client can submit her *request*, which consists in: (i) a DIG expression of the required resource features; (ii) geographical coordinates of the current device location; (iii) maximum acceptable distance for service/resource fruition.

When a request is received, the server performs the following processing steps. 1. Resources referring to the same ontology are extracted from the registry. 2. A location-based pre-filter excludes resources outside the maximum range w.r.t. the request, as explained in the subsection below. 3. The reasoning engine computes the semantic distance between request and each resource in range. 4. Results of semantic matchmaking are transferred to the utility function calculation module, which computes the final ranking according to the scoring functions reported in the next subsection. 5. Finally, the ranked list of best resource records is sent back to the client in a DIG reply.

A. Location-based resource filtering

Semantic-based matchmaking should be extended to take location into account, so as to provide an overall match degree that best suits the user needs in her current situation. Research in logic-based matchmaking has achieved some degree of success in extending useful inference services to DLs with *concrete domains* (*datatype properties* in Semantic Web words) [2], nevertheless these results are hardly transferred to mobile scenarios due to performance limitations. A different approach to the multi-attribute resource ranking problem is based on *utility functions*, a.k.a. *Score Combination Functions (SCF).* It consists in combining semantic-based match metrics with other partial scores computed from quantitative –in our case location-dependent– resource attributes.

In general, if a request and available resources are characterized by *m* attributes, the problem is to find a ranking of the set *R* of supplied resources according to the request $d = (d_1, d_2, \ldots, d_m)$. For each resource $r_i = (r_{i,1}, r_{i,2}, \ldots, r_{i,m}) \in R, 1 \le i \le |R|$, a set of *local scores* $s_{i,j}, 1 \le j \le m$ is computed as $s_{i,j} = f_j(d_j, r_{i,j})$. Then the *overall score* s_i for r_i is obtained by applying an SCF *f*, that is $s_i = f(s_{i,1}, s_{i,2}, \ldots, s_{i,m})$. Resources are so sorted and ranking is induced by the SCF.

The framework devised in this paper integrates a *semantic score* f_{ss} and a *geographic score* f_{gs}, combined by the SCF f_{sc}. The operating principle is illustrated in Figure 2: a circular area is identified, centered in the user's position; the service provider will only return resources located in it. The user request contains a *(latitude, longitude)* pair of geographical coordinates for current device location along with a maximum range *R*. In the same way, each available resource collected by the provider is endowed with its coordinates. Distance *d* is computed between the user and the resource. If *d* > *R*, the resource is excluded, otherwise it is admitted to next processing stage.

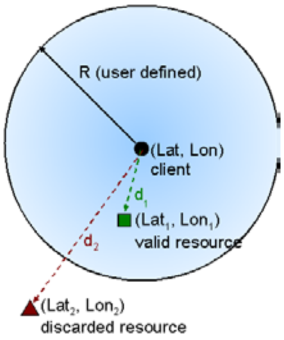


Figure 2. Location-based resource pre-filtering.

The semantic score is computed as:

$$f_{ss}(r,s) = \frac{s\_match(r,s)}{\max(s\_match)}$$

where $s\_match(r,s)$ is the semantic match distance from request *r* to resource *s* (computed by means of the inference services explained before), while $\max(s\_match) \doteq s\_match(r,\top)$ is the maximum semantic distance, which depends on axioms in the reference domain ontology. Hence, $f_{ss} \in [0,1]$ and lower values are better.

The second score involves the physical distance:

$$f_{gs}(d) = \frac{d}{R}$$

Also $f_{gs} \in [0,1]$ and lower values are preferable. It should be noticed that, in both local scoring functions, denominators are maximum values directly depending on the specific user request. They may change across different resource retrieval sessions, but correctly rank resources w.r.t. the request within the same session.

Finally, the SCF is defined as:

$$f_{sc}(d,S) = 100 \cdot [1 - (f_{gs} + \varepsilon)^{\alpha \frac{R}{\beta}} \cdot (f_{ss} + \gamma)^{1-\alpha}]$$

It is a monotonic function providing a consistent resource ranking, and it converts results to a more user-friendly scale where higher outcomes represent better results. A *tuning* phase can be performed to determine parameter values following requirements of a specific discovery application. In detail, $\alpha \in [0,1]$ weighs the relevance of semantic and geographic factors, respectively. With $\alpha \to 0$ we privilege the semantic score, whereas with $\alpha \to 1$ the geographic one is made more significant. The exponent of the geographic factor is multiplied by $\frac{R}{\beta}$. This is because, when the maximum search range *R* grows, distance should reasonably become a more selective attribute, giving more relevance to resources in the user's immediate proximity. The coefficient β regulates the curve decay, as shown in Figure 3 for different values of β and $\alpha = 0.5$, $\varepsilon = 0$, $d = 30$ km.

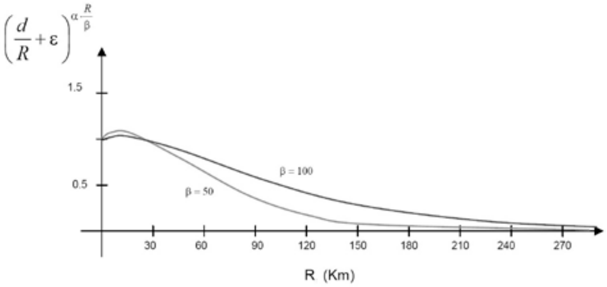


Figure 3. Geographic score contribution w.r.t. range R

Parameters $\varepsilon \in [0,1]$ and $\gamma \in [0,1]$ control the outcome in case of either semantic or geographic *full match*. As explained in Section II, semantic full match occurs when all features in the request are satisfied by the resource. Geographic full match occurs when the user is located exactly in the same place of resource she is looking for. Both cases are desirable but very unlikely in practical scenarios. Hence, in the model adopted for system evaluation we could pose $\varepsilon = \gamma = 0$:

$$f_{sc}(d,S) = 100 \cdot [1 - (f_{gs})^{\alpha \frac{R}{\beta}} \cdot (f_{ss})^{1-\alpha}]$$

This means that full matches will always be shown at the top of the result list, since either $f_{gs} = 0$ or $f_{ss} = 0$ implies $f_{sc} = 100$ regardless of the other factors.

B. *Design and development methodology*

Mobile computing devices are very heterogeneous in terms of screen size, input methods, computational and communication capabilities and operating systems. Furthermore, Human-Computer Interaction (HCI) design should endorse the peculiarities of handheld devices. Unlike their desktop counterparts, mobile applications are characterized by a *bursting* usage pattern, *i.e.*, with frequent and short sessions. Hence, a mobile GUI must be designed so that users can satisfy their needs in a quick and straightforward way. A task-oriented and consistent look and feel is required, leveraging familiar metaphors, which most users are accustomed to. Finally, software design must take into account the inherent constraints of mobile ad-hoc networks: from the application perspective, the most important issues are unpredictable disconnections and low data rates.

For a greater compatibility with various mobile platforms, our client tool was developed using Java Micro Edition (ME) technology. The Java Mobile Information Device Profile 2.0 (JSR 37, JSR 118, available at http://www.oracle.com/technetwork/java/index-jsp-138820.html) was selected. All UI elements are accessible either through the keyboard/keypad or the touchscreen of the mobile device.

The MVC (Model-View-Controller) pattern was adopted in user interface design. This was important for the management and presentation of semantic-based data, which have an intrinsically complex data model. The GUI was entirely based on SVG (Scalable Vector Graphics), using the Scalable 2D Vector Graphics for Java ME (JSR 226, available at http://jcp.org/en/jsr/detail?id=226).

The application exploits the Java Location API (JSR 179, available at http://jcp.org/en/jsr/detail?id=179) to allow location-based service/resource provisioning. It provides a unified API to interact with all *location providers –i.e.*, real-time positioning technologies– available on the device. These may include a GPS (Global Positioning System) receiver and the mobile phone network itself (cell-based positioning).

The proposed tool supports a subset of the DIG 1.1 interface extended for MaMaS-tng reasoner (see the MatchMaking Service, available at http://sisinfab.poliba.it/MAMAS-tng/). A lightweight implementation of the client-side DIG interface has been developed in Java. A specialized library was designed to manipulate Knowledge Bases (KBs). In order to minimize runtime memory usage, *kXML* (kXML 2, available at http://kxml.sourceforge.net/) Java streaming XML parser was adopted, which implements the open standard XML Pull API. Streaming parsers do not build an in-memory syntax tree model. They are also suitable for XML data incoming from network connections, since parsing can be pipelined with the incoming input.

IV. CASE STUDY

Functional and non-functional features of the proposed system are motivated within a concrete case study in the cultural heritage tourism sector.

Let us model the discovery problem as follows. *Jack is in Bari for business. He is keen on ancient architecture and after his last meeting, he is near the old town centre with some spare time. He had never been in Bari before and he knows very little about the city. Being interested in medieval art and particularly in churches, he would like to visit interesting places near his current location. Under GPRS/UMTS or Wi-Fi coverage, his GPS-enabled smartphone can connect to a service/resource provider, in order to search for interesting items in the area.* The service provider keeps track of semantic annotations of touristic points of interest in Apulia region along with their position coordinates. The mobile application assists the user in the discovery process through the following three main tasks (depicted in Figure 4).

Ontology management. *Firstly, Jack selects cultural heritage as the resource category he is interested in.* Different domain ontologies are used to describe general resource classes (*e.g.*, accommodation, cultural heritage, movie/theatre shows). At application startup, a selection screen is shown (Figure 5), with a list of managed ontologies. Each Ontology is labeled by a Universally Unique IDentifier (OUUID), which allows an early agreement between user and provider. As explained in [14], this simple identification mechanism borrowed from the Bluetooth Service Discovery Protocol allows to perform a quick match between the ontologies managed by the user and the provider also in case of mobile ad-hoc connections where users and providers are interconnected via wireless links (such as Bluetooth, 802.11, ZigBee and so on) and where a dependable Web link is unavailable. Anyway, in case the user cannot locally manage the chosen resource category, he can download the reference ontology either from near hosts or from the Web (when possible) exploiting the OUUID as reference identifier.

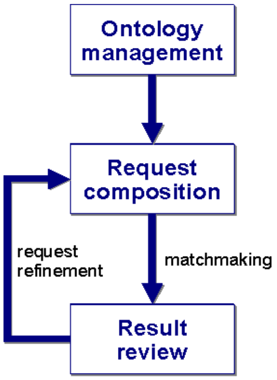
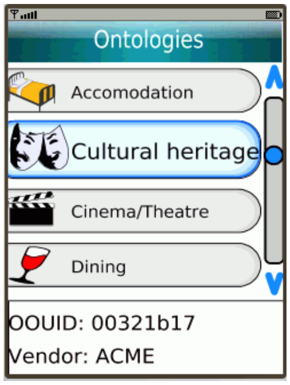


Figure 4. Tasks of the mobile client.

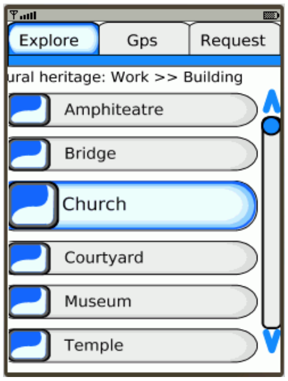


OOUID: 00321b17
Vendor: ACME

Figure 5. Ontology selection screen

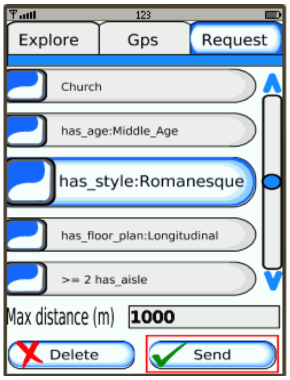
Semantic request composition. *Jack composes his semantic-based request through a fully visual form. He browses resource features modeled in the domain ontology* (partially reported in Table I for the sake of brevity) *and selects desired characteristics, without actually seeing anything of the underlying DL-based formalism. Then he submits his request.*

Figure 6 shows the ontology browsing screen. A scrollable list shows the current *focus* in the classification induced by terminological definitions and subsumptions. Directional keys of mobile device or swipe gestures on the touchscreen are used to browse the taxonomy by expanding an item or going back one level. Above the list, a *breadcrumb* control is displayed, so that the user can orient himself even in deeper ontologies.



ural heritage: Work >> Building

Figure 6. Ontology browsing screen



Church
has_age:Middle_Age
has_style:Romanesque
has_floor_plan:Longitudinal
>= 2 has_aisle
Max distance (m) 1000

Figure 7. Request confirmation screen

TABLE I. EXCERPT OF AXIOMS IN THE CASE STUDY ONTOLOGY

$\text{AD} \sqsubseteq \text{Age}$	$\text{BC} \sqsubseteq \text{Age}$	$\text{Middle\_Age} \sqsubseteq \text{AD}$
$\text{Centralized} \sqsubseteq \text{Floor\_Plan}$	$\text{Longitudinal} \sqsubseteq \text{Floor\_Plan}$	$\text{Quadrangular} \sqsubseteq \text{Floor\_Plan}$
$\text{Square} \sqsubseteq \text{Quadrangular}$	$\text{Byzantine} \sqsubseteq \text{Style}$	$\text{Romanesque} \sqsubseteq \text{Style}$
$\text{Gothic} \sqsubseteq \text{Style}$	$\text{Baroque} \sqsubseteq \text{Style}$	$\text{Portal} \sqsubseteq \text{Architectural Element}$
$\text{Cathedra} \sqsubseteq \text{Architectural\_Element}$	$\text{Aisle} \sqsubseteq \text{Architectural\_Element}$	$\text{Altar} \sqsubseteq \text{Architectural\_Element}$
$\text{Pulpit} \sqsubseteq \text{Architectural\_Element}$	$\text{Crypt} \sqsubseteq \text{Architectural\_Element}$	$\text{Apse} \sqsubseteq \text{Architectural\_Element}$
$\text{Window} \sqsubseteq \text{ArchitecturalElement}$	$\text{Single\_Light} \sqsubseteq \text{Window}$	$\text{Double\_Light} \sqsubseteq \text{Window}$
$\text{Triple \_Light} \sqsubseteq \text{Window}$	$\text{Religious} \sqsubseteq \text{Destination}$	$\text{Private} \sqsubseteq \text{Destination}$
$\text{Public} \sqsubseteq \text{Destination}$	$\text{Private} \sqsubseteq \neg\text{Public}$	$\text{Private} \sqsubseteq \neg\text{Religious}$
$\text{Building} \sqsubseteq \exists\,\text{has\_age} \sqcap \exists\,\text{has\_floor\_plan} \sqcap \exists\,\text{has\_style}$		
$\text{Residence} \sqsubseteq \text{Building} \sqcap \exists\,\text{Destination} \sqcap \forall\,\text{Destination.Private}$		
$\text{Church} \sqsubseteq \text{Building} \sqcap \exists\,\text{Destination} \sqcap \forall\,\text{Destination.Religious} \sqcap \exists\,\text{has\_altar} \sqcap \forall\,\text{has\_altar.Altar}$		
$\text{Castle} \sqsubseteq \text{Residence}$		

The tabs on top of the screen allow user to switch from the Explore screen to the Request confirmation screen (Figure 7). There the user can remove previously selected features. Eventually, he specifies a retrieval diameter *R* and submits his request. Current prototype expresses the threshold in terms of distance, but a more intuitive indication clarifying if the user is on foot (possibly also specifying the terrain characteristics) or if he moves by car is also possible.

Jack would like to visit a Romanesque Middle Age church with longitudinal floor plan and two aisles. W.r.t. the cultural heritage ontology, the request can be formally expressed as:

R: $\text{Church} \sqcap \forall \text{has\_age.Middle\_Age} \sqcap \forall \text{has\_floor\_plan.Longitudinal} \sqcap \geq 2\ \text{has\_aisle} \sqcap \forall \text{has\_style.Romanesque}$

It can be noticed that requests are formulated as DL conjunctive queries. Each conjunct is a requested resource feature; it can be an atomic concept selected from the ontology, a universal quantifier or an unqualified number restriction on roles. The GUI masks this level of complexity from the user, allowing him to simply browse lists of features and select the desired ones: translation into DL expression is automated, taking into account the concept structure and relationships in the reference ontology,

The communication module was designed as a finite state machine to precisely retain knowledge about the progress of client-server interaction. By doing so, only failed operations are actually repeated, thus improving efficiency from both time and energy standpoints.

Results review and query refinement. *The server processes the request as explained in Section 3.* Let us consider the following resources in the provider KB:

S1: Basilica of St. Nicholas, Bari (distance from user: d = 0.9 km). A Romanesque Middle Age church, with longitudinal floor plant, an apse, two aisles, three portals and two towers. Other notable elements are its crypt, altar, cathedra and Baroque ceiling. W.r.t. domain ontology, this is expressed as:

$\text{Church} \sqcap =2\ \text{has\_aisle} \sqcap \forall \text{has\_age.Middle\_Age} \sqcap \forall \text{has\_style.Romanesque} \sqcap =1\ \text{has\_apse} \sqcap =3\ \text{has\_portal} \sqcap =1\ \text{has\_crypt} \sqcap =1\ \text{has\_altar} \sqcap =2\ \text{has\_tower} \sqcap =1\ \text{has\_cathedra} \sqcap \exists\ \text{ceiling\_style} \sqcap \forall\ \text{ceiling\_style.Baroque} \sqcap \forall \text{has\_floor\_plan.Longitudinal}$

S2: Norman-Hohenstaufen Castle, Bari (d = 0.57 km). It is described as a Middle Age castle, with Byzantine architectural style and a quadrangular plan with four towers.

$\text{Castle} \sqcap \forall\ \text{has\_floor\_plan.Quadrangular} \sqcap =4\ \text{has\_tower} \sqcap \forall \text{has\_style.Byzantine} \sqcap \forall \text{has\_age.Middle\_Age}$

S3: Church of St. Scholastica (d = 1.3 km). It is described as a Romanesque Middle Age church, with longitudinal floor plan, three aisles, an apse and a tower.

$\text{Church} \sqcap \forall\ \text{has\_style.Romanesque} \sqcap \forall \text{has\_age.Middle\_Age} \sqcap \forall \text{has\_floor\_plan.Longitudinal} \sqcap =3\ \text{has\_aisle} \sqcap =1\ \text{has\_tower} \sqcap =1\ \text{has\_apse}$

S4: Church of St. Mark of the Venetians, Bari (d = 0.65 km). It is described as a Romanesque Middle Age church with two single-light windows and a tower.

$\text{Church} \sqcap \forall\ \text{has\_style.Romanesque} \sqcap \forall\ \text{has\_window.Single\_Light} \sqcap =2\ \text{has\_window} \sqcap \forall \text{has\_age.Middle\_Age} \sqcap =1\ \text{has\_tower}$

Table II reports matchmaking results for the above example. **S3** is discarded in the location-based pre-filtering, as its distance from the user exceeds the limit, even though it would result in a full match. **S1** is a full match with the request, because it explicitly satisfies all user requirements. On the other hand, **S4** is described just as Romanesque Middle Age church, therefore due to OWA it is not specified whether it has a longitudinal floor plan with aisles or not: these characteristics become part of the *Hypothesis* computed through CAP. Finally, **S2** produces a partial match with user request, since it refers to a castle: this concept is incompatible with user request, so it forms the *Give Up* feature computed through CCP.

TABLE II. MATCHMAKING RESULTS

Supply	**Match type**	**s_match [max =54]**	**Outcome**	**Score [α=0.5,β=1, γ=0.014, ε=0]**
S1: Basilica of St. Nicholas	Full	0	Hypothesis H: $\top$ Bonus B: $=1\ \text{has\_apse} \sqcap =3\ \text{has\_portal} \sqcap =1\ \text{has\_crypt} \sqcap =1\ \text{has\_altar} \sqcap =2\ \text{has\_tower} \sqcap =1\ \text{has\_cathedra} \sqcap \exists\ \text{ceiling\_style} \sqcap \forall\ \text{ceiling style.Baroque}$	88.8
S4: Church of St. Mark	Potential	3	Hypothesis H: $\geq 2\ \text{has\_aisle} \sqcap \forall\ \text{has\_floor\_plan. Longitudinal}$ Bonus B: $=1\ \text{has\_tower} \sqcap =2\ \text{has\_window} \sqcap \forall \text{has\_window.Single\_Light}$	78.3
S2: Norman-Hohenstaufen Castle	Partial	11	Give up G: Church Keep K: $\text{Building} \sqcap \forall \text{has\_age.Middle\_age}$ Hypothesis H: $\forall \text{has\_floor\_plan.Longitudinal} \sqcap \geq 2\ \text{has\_aisle} \sqcap \forall \text{has\_style.Romanesque}$ Bonus B: $=4\ \text{has\_tower} \sqcap \forall \text{has\_style.Byzantine}$	64.8
S3: Church of St. Scholastica	N.A.	N.A.	Discarded due to distance	N.A.

Overall scores of advertised resources are finally computed. The result screen is reported in Figure 8: retrieved resources are listed, best matching first. When the user selects a resource, its picture is shown as in Figure 9 in addition to its address, distance from the user and semantically relevant properties contributing to the outcome.

If Jack is not satisfied with results, he can refine his request and submit it again. The user can go back to the ontology browsing screen to modify the request. Furthermore, he can select some elements of the *Bonus* (respectively *Give Up*) list in the result screen and they will be added to (resp. removed from) the request.

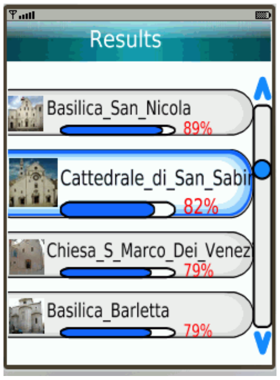
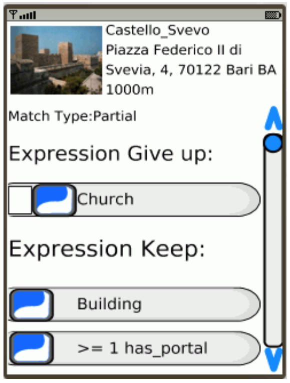


Figure 8. Displayed results



Castello_Svevo
Piazza Federico II di Svevia, 4, 70122 Bari BA
1000m

Match Type:Partial

Expression Give up:

□ Church

Expression Keep:

Building

>= 1 has_portal

Figure 9. Result details screen

V. SYSTEM EVALUATION

Common issues rising from the integration of Semantic Web approaches with ubiquitous computing scenarios were evidenced in [17]. Let us take them as a check-list and evaluate our proposal against it.

A. Simple architectures lack intelligence of Semantic Web technologies. The current proposal allows mobile devices equipped with commonly available technologies to fully exploit semantic-based resource discovery. Ideas and technologies devised for resource retrieval in the Semantic Web were adapted with a satisfactory success, through careful selection of features and optimization of implementations.

B. Semantic Web architectures use devices with a secondary, passive role. In our prototype the client has a key role and it does not only act as a GUI for request composition via ontology browsing. It also enables: location determination; interaction with a state-of-the-art DIG-based reasoning engine; interactive visualization of discovery results for query refinement.

C. Semantic Web architectures rely on a central component that must be deployed and configured beforehand for each specific scenario. The proposed system prototype still relies on a centralized server for resource matchmaking. Future work aims at building a fully mobile peer-to-peer architecture. A major step is to design and implement embedded DL reasoners with acceptable performance: early results have been achieved in this concern [14].

D. Most architectures do not use the Web communication model, essentially HTTP. For communication we only use DIG, a standard based on the HTTP POST and an XML-based concept language. Such a choice allows –among other things– to cope with scalability issues: particularly, the interaction model is borrowed from the Web experience in order to grant an acceptable behavior also in presence of large amounts of exchanged data.

E. Devices are not first-class actors in the environment with autonomy, context-awareness and reactiveness. Though the typical usage scenario for our current prototype is user-driven, it shows how a non-technical user can fully leverage Semantic Web technologies via her personal mobile device to discover interesting resources in her surroundings.

VI. CONCLUSION AND FUTURE WORK

A framework has been presented for semantic-enabled resource discovery in ubiquitous computing. It has been implemented in a visual mobile DSS able to retrieve resources/services through a fully dynamic wireless infrastructure, without relying on support facilities provided by wired information systems. It recognizes via GPS the user location and grades matchmaking outcomes according to proximity criteria. Future work aims at simplifying the complexity of matchmaker module claiming for optimization and rationalization of the reasoner structure, in order to improve performance and scalability and to allow its integration into mobile computing devices and systems. Furthermore, the application user interface has to be enhanced and redesigned to be even more friendly for non-expert users. We are investigating a new approach directly and automatically browsing the DBpedia KB.

REFERENCES

[1] Langegger, A. and Wöß W. “Product finding on the semantic web: A search agent supporting products with limited availability”. *International Journal of Web Information Systems*, Vol. 3, No. 1/2, Emerald Group Publishing Limited, 2007, pp. 61-88.

[2] Baader, F., Calvanese, D., Mc Guinness, D., Nardi, D., and Patel-Schneider, P. *The Description Logic Handbook*. Cambridge University Press, New York, 2002.

[3] Watson, R., Akselsen, S., Monod, E., and Pitt, L. “The Open Tourism Consortium: Laying The Foundations for the Future of Tourism.” *European Management Journal*, Vol. 22, No. 3, 2004, pp. 315–326.

[4] Colucci, S., Di Noia, T., Pinto, A., Ragone, A., Ruta, M., and Tinelli, E. “A Non-Monotonic Approach to Semantic Matchmaking and Request Refinement in E-Marketplaces”. *International Journal of Electronic Commerce*, Vol. 12, No. 2, 2007, pp. 127-154.

[5] McGuinness D.L. and van Harmelen F., editor. “OWL Web Ontology Language, W3C Recommendation, 2004. http://www.w3.org/TR/owlfeatures/”. Accessed 14 October 2008.

[6] Bechhofer, S., Möller, R., and Crowther, P. “The DIG Description Logic Interface.” In: *Proceedings of the 16th International Workshop on Description Logics (DL’03)*. Vol. 81 of *CEUR Workshop Proceedings*, 2003.

[7] Li, L. and Horrocks, I. “A Software Framework for Matchmaking Based on Semantic Web Technology. *International Journal of Electronic Commerce*, Vol. 8(4), 2004, pp. 39–60.

[8] Di Noia, T., Di Sciascio, E., Donini, F.M., and Mongiello, M. “Abductive matchmaking using description logics.” In: *Proceedings of Eighteenth International Joint Conference on Artificial Intelligence IJCAI-03*, Acapulco, Mexico, MK, Vol. 18, 2003, pp. 337-342.

[9] Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F.M., and Ragone, A. “Knowledge Elicitation for Query Refinement in a Semantic-Enabled E-Marketplace”. In *7th International Conference on Electronic Commerce ICEC 05*, ACM Press, 2005, pp. 685-691.

[10] Noppens, O., Luther, M., Liebig, T., Wagner, M., and Paolucci, M.. “Ontology supported Preference Handling for Mobile Music Selection.” In: *Proceedings of the Multidisciplinary Workshop on Advances in Preference Handling*, Riva del Garda, Italy, 2006.

[11] Wilson, M., Russell, A., Smith, D., Owens, A., and Schraefel, M. “mSpace Mobile: A Mobile Application for the Semantic Web.” In: *Proceedings of the End User Semantic Web Workshop at ISWC 2005*, 2005.

[12] Sinner, A. and Kleemann, T. “KRHyper - In Your Pocket.” In: *Proc. of 20th International Conference on Automated Deduction (CADE-20)*, 2005, pp. 452–457.

[13] Kleemann, T. and Sinner, A. “User Profiles and Matchmaking on Mobile Phones.” In Bartenstein, O., ed.: *Proc. of 16th International Conference on Applications of Declarative Programming and Knowledge Management INAP2005*, 2005.

[14] Ruta, M., Di Noia, T., Di Sciascio, E., and Scioscia, F. “Abduction and Contraction for Semantic-based Mobile Dating in P2P Environments.” In: *Proc. of 6th IEEE/WIC/ACM International Conference on Web Intelligence WI08*, IEEE, 2008, pp. 626–632.

[15] Bizer, C., Heath, T. , Idehen, K., Berners-Lee, T. “Linked data on the web”, In: *Proceeding of the 17th international conference on World Wide Web*, 2008, pp. 1265-1266, ACM.

[16] Ruta, M., Di Noia, T., Di Sciascio, E., and Donini, F.M.. “Semantic based collaborative p2p in ubiquitous computing.” *Web Intelligence and Agent Systems*, Vol. 5, No. 4, 2007, pp. 375–391.

[17] Vazquez, J.I., López de Ipiña, D., and Sedano, I. “SoaM: A Web-powered Architecture for Designing and Deploying Pervasive Semantic Devices”. *International Journal of Web Information Systems*, Vol. 2, No. 3/4, Emerald Group Publishing Limited, 2006, pp. 212-224.

Exploiting WordNet glosses to disambiguate nouns through verbs

Donato Barbagallo, Leonardo Bruni, and Chiara Francalanci
Department of Electronics and Information
Politecnico di Milano
Milano, Italy
{barbagallo,bruni,francala}@elet.polimi.it

Abstract—**This paper presents an unsupervised graph-based algorithm for word sense disambiguation based on WordNet glosses. The algorithm exploits the contributions of verbs in identifying the correct senses of nouns. Due to the complexity of WordNet's semantic network, we have defined disambiguation as a similarity optimization problem and solved it through a genetic algorithm. Testing compares the performance of our algorithm with that of a traditional method based on Wu-Palmer similarity measure. Our approach shows an overall precision of about 68% and a statistically significant average increase of precision of about 3% with respect to the traditional algorithm.**

Keywords - word sense disambiguation; WordNet; genetic algorithm.

I. INTRODUCTION

Word Sense Disambiguation (WSD) is the ability of identifying the meaning of words in a given sentence. It represents a fundamental research problem in Natural Language Processing with many practical applications, such as search engines, information retrieval, and sentiment analysis.

WSD has made a considerable progress in the last few years and can now obtain good results through supervised algorithms. Though results can be very precise, the literature recognizes the high costs and strong feasibility limits of these techniques, due to their need for context-dependent annotated corpora [6]. On the other hand, unsupervised techniques can also be applied to WSD. In this field, the term unsupervised is usually referred to techniques that are not necessarily knowledge-free, since some kind of knowledge base, i.e. dictionaries or computational lexicons, is needed [2]. These knowledge bases usually provide a context-independent sense inventory and relations among senses, which can be exploited to perform WSD. Though the literature tends to recognize that supervised methods usually outperform unsupervised ones, from a cost-benefit analysis point of view it can be still more convenient to invest and develop unsupervised methods. Indeed, in some applications of WSD, such as information retrieval, perfect word sense information would be of limited utility [5].

The literature has often combined unsupervised methods based on semantic networks such as WordNet [12] with the so-called similarity measures. These measures assume that two words are similar when they appear in a similar context, e.g., in the same sentence of paragraph, and contexts are similar when they contain similar words [28]. According to these assumptions, word senses whose definitions have the highest score of similarity are assumed to be the correct ones [27][28].

In this paper, we present a new unsupervised method to disambiguate nouns based on WordNet and on the concept of similarity. The innovative aspect of this method is that it is able to exploit WordNet glosses and verbs and create the link between nouns and verbs sub-graphs, outperforming traditional approaches based only on nouns. We compare the performance of our algorithm with a classical disambiguation approach based on nouns and the Wu-Palmer similarity measure.

There are many techniques in literature to solve the similarity optimization problem. This work uses a genetic algorithm to solve the similarity problem. Although genetic algorithms are global search heuristics and their results are not guaranteed to be optimal solutions, they are also known to outperform other optimization techniques when the problem space is large, as in the case of a semantic network made of hundreds of thousands of words. Moreover, genetic algorithms perform better when the solution space is discontinuous and include multiple local optima [30].

The remainder of this paper is structured as follows. Section II explains the background knowledge behind our algorithm. Section III presents the algorithms, Section IV describes the experiments and discusses the results. Section V presents related research and highlights the innovative aspects of our work. Finally, Section VI shows some conclusion and presents future research directions.

II. METHODOLOGY AND TECHNOLOGY RESOURCES

In this section, the methodology and the technology resources are briefly described.

A. Genetic Algorithm

Evolutionary computation techniques [11] make use of Darwin's evolutionary principles and translate them into heuristic algorithms that can be used to search for optimal solutions to a problem. In a search algorithm, the objective is to find the best possible solution in a fixed amount of time. When the search space grows in size, an exhaustive search becomes quickly unfeasible. The key aspect distinguishing an evolutionary search algorithm from more traditional heuristic algorithms is that it is *population-based*. Through the adaptation of successive generations of a large number of individuals, an evolutionary algorithm performs an efficient search.

Genetic algorithms are a particular class of evolutionary algorithms. In a genetic algorithm, a potential solution is represented by a *chromosome*, usually encoded as an array of bits or characters. A single bit or a set of bits coding part of the solution is called *gene*. In turn, an *allele* is one of the possible instances of the gene.

The first population is typically randomly generated. Then a measure of goodness (necessarily domain dependent) is computed for each chromosome. Guided by this quantitative information, together with a set of genetic operators like crossover and mutation, genetic algorithms move from one population of chromosomes to a new population. Typically, the evolution terminates when either a fixed number of generations has been created or the fitness value of a chromosome reaches a target threshold.

Genetic algorithms have been used for many applications like optimization, classification, prediction, economy, ecology and automatic programming.

B. *WordNet*

WordNet [12] is a freely available lexical database for the English language that organizes nouns, verbs, adjectives and adverbs into hierarchies of *synonym sets* or *synsets*. Each synset groups words with a unique meaning and it has a gloss that describes the concept that it represents. For example, the synset composed by the words {apartment, flat} represents the concept defined by the gloss "a suite of rooms usually on one floor of an apartment house". Many glosses are extended with the addition of some examples of usages of the concept that they describe.

WordNet is organized as a network of concepts linked by semantic relations, like *hypernym*, *hyponym*, *meronym*, *holonym*, and *antonym*. However, these relations do not cross part of speech boundaries. Thus, semantic relations are tied to a particular part of speech, creating different and separate sub-graphs for nouns, verbs, adjectives and adverbs. In our experiments we use WordNet 3.0 and we focus on the hypernym hierarchy that represents the most complete set of relations.

III. The WSD System

We follow two strategies to perform the disambiguation of nouns. The first, called *base algorithm*, relies only on the information carried by the nouns in a sentence. The second, called *enhanced algorithm*, aims at improving the results exploiting also the information that can be extracted from verbs through disambiguated glosses.

A. *Base algorithm*

The basic idea of our WSD system, also exploited in [10], relies on the assumption that terms that appear in the same sentence tend to be semantically similar. The genetic algorithm is used to find a set of senses that maximizes the similarity between the terms to be disambiguated. Because both the number of possible senses for each word and the cardinality of the set of words to disambiguate can be large the search space become huge. Thus genetic algorithms are a suitable solution for this kind of problem.

Similarity is a widely used concept. According to Budanitsky and Hirst [1], it is possible to make a distinction between semantic similarity and relatedness: semantic similarity is a special kind of relatedness between two words and denotes the degree of semantic association between them. Measures of relatedness can be made across part of speech boundaries and are not tied to the is-a relation. However, for the purpose of this paper, we refer to both kinds of measure with the term similarity. Many similarity measures have been proposed, such as information content [4], Lin [7], Jiang-Conrath [3], Banarjee-Pedersen [9], Wu-Palmer [8]. The whole set of cited measures has been compared through some preliminary experiments showing that Wu-Palmer can be considered the measure providing the best results.

Wu-Palmer defines the similarity of two concepts by measuring how closely they are related in the hierarchy, i.e., the similarity measure between a pair of concepts *c1* and *c2* is:

$$sim_{WP}(c1, c2) = (2 * N3)/(N1 + N2 + 2 * N3) \qquad (1)$$

where N1 is the number of nodes on the path from c1 to c3 (the least common superconcept of c1 and c2), N2 is the number of nodes on the path from c2 to c3, and N3 is the number of nodes on the path from c3 to the root of the hierarchy.

Table I shows some examples of measures between pairs of nouns computed by (1), where with city#1 we mean the first sense of “city”, with animal#1 we mean the first sense of “animal” and so on. The first sense of “turkey” is “large gallinaceous bird with fan-shaped tail; widely domesticated for food” and its second sense is “a Eurasian republic in Asia Minor and the Balkans”. As we can expect, the concept of “city” intended in its first sense, i.e., “a large and densely populated urban area”, is more similar with the second sense of “turkey” than with the first sense. Analogously, the concept of “animal” intended in its first sense, i.e., “a living organism characterized by voluntary movement”, is more similar with the first sense of “turkey” than with the second sense.

TABLE I. SIMILARITY BY WU-PALMER’S MEASURE

	turkey#1	turkey#2
city#1	0.20	0.75
animal#1	0.67	0.29

In our system the similarity measure presented above represents the core of the fitness function of the genetic algorithm. Each solution is represented by a chromosome that is encoded as a sequence of positive integer numbers. Each gene of a chromosome is a possible sense of a term. The fitness value for each chromosome is computed as follows:

$$F = \sum_i \sum_{j=i+1} sim(s(w_i), s(w_j)) \qquad (2)$$

where w_i and w_j are two terms, $s(w_i)$ and $s(w_j)$ are the candidate senses of w_i and w_j.

The similarity measure used in (2) is slightly different from the measure in (1). In order to perform better on general documents, the original value is weighted by the frequency of the words' sense, because in general context, words tend to assume their more frequent meaning. In WordNet, word senses are ordered by their frequency of use, i.e. the most frequent senses are indicated with lower ordinal numbers. So we define the new similarity measure as:

$$sim\left(s(w_i), s(w_j)\right) = \left(\frac{1}{n_i} + \frac{1}{n_j}\right) * sim_{WP}\left(s(w_i), s(w_j)\right) (3)$$

where n_i and n_j denote the ordinal number of $s(w_i)$ and $s(w_j)$ as reported by WordNet.

B. Enhanced algorithm

We have observed that the information carried by the nouns may not be enough. For example, if we want to disambiguate a sentence like "I ate a tasty turkey for Christmas" using the Base Algorithm, the set of nouns used for the disambiguation is composed by the words {turkey, Christmas}. Just considering the couple it is not clear whether the noun *turkey* has to be interpreted as fowl or Eurasian republic. On the other hand, if the verb *eat* is added to the set, it becomes clear that the former is the right meaning, because the verb carries contextual information.

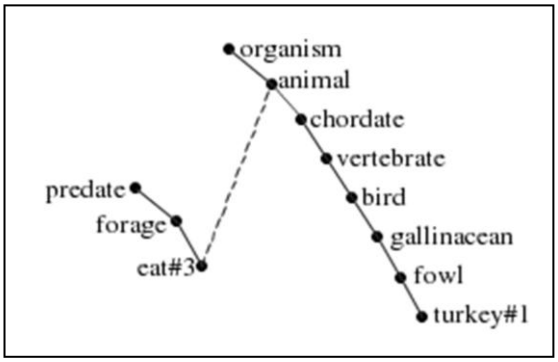


Figure 1. Example of a new verb-noun relation in WordNet

Similarity measures can be applied only to pairs of words of the same part of speech. To deal with this limit we make use of an additional resource [13] where word forms from the definitions (called *glosses*) in WordNet's synsets are manually linked to the context-appropriate sense in WordNet.

In order to take into account this new kind of information we extended the size of the chromosome defined in the previous algorithm. For each sense of each verb found in the sentence of which we want to disambiguate the terms we extract a noun from its annotated gloss and then a new gene is added to the chromosome whose possible value is only the sense with which it was tagged in the gloss. We provide a simple example using the set {{turkey, Christmas},{eat}}. The base form of the chromosome will have two genes since there are two nouns in the set. The verb *eat* in WordNet has six senses, so we add six "monosemic" genes to the chromosome, each one representing one noun extracted from the gloss of each sense of the verb. The new chromosome is shown in Table II. Figure 1 graphically represents the example and shows how easily it is possible to create a relation (dashed line) between a noun and a verb, while unbroken lines represent hypernym hierarchies for the verb eat (synset #3) and the noun turkey (synset #1).

TABLE II. EXAMPLE OF CHROMOSOME

Gene no.	Noun	Alleles
1	turkey	1-5
2	christmas	1-2
3	solid_food	1
4	meal	1
5	animal	1
6	way	1
7	resource	1
8	action	4

When one of the new genes is involved in the computation of the similarity value, the frequency with which is weighted is the frequency of the verb that has generated the gene.

IV. EVALUATION AND DISCUSSION

Experiments are performed using the JGAP [29] library to implement the genetic algorithm. Due to the intrinsic heuristic nature of genetic algorithms, we performed several tests with different settings of the parameters of the genetic operators. These tests have highlighted that result deltas are irrelevant. In this section, we present the results obtained by applying the default configuration of genetic operators as provided by the genetic algorithm implementation in JGAP. Similar tests executed by varying the population size have highlighted also the fact that precision does not increase significantly when the population size overgrows ten chromosomes. Given that WordNet word senses are ordered by frequency of use, the first ten senses are sufficient to cover the common usage of words.

We have performed a sentence by sentence analysis for two main motivations: *(i)* in our case, the disambiguation process is part of a larger project on sentiment analysis (cf. [31]) considering short sentences, such as tweets or blog posts, and *(ii)* we found out that the number of words to disambiguate and precision are uncorrelated, as shown later in this section.

Algorithm performance is measured in terms of precision and recall. Following [32], precision is defined as the number of correct disambiguated senses divided by the total number of answers reported; recall is defined as the number of correct disambiguated senses divided by the total number of senses. Since our methods can assign a sense for every word, precision equals recall.

The results of the experiments are evaluated on SemCor Corpus, the sense-tagged version of the Brown Corpus, by automatically comparing the sense-tags in SemCor with those computed by our algorithms. We carried out experiments over 19 randomly selected SemCor files (br-a02, c01, e04, e27, f10, f22, f43, g18, g19, g28, h18, j04, j12, j20, j57, j70, k04, l18, r05).

Because of the heuristic nature of genetic algorithms we run each test ten times in order to have an empirical assessment of the variability of the results.

Table III shows the comparison statistics between our base and enhanced algorithms. It is worth noting how enhanced algorithm outperforms base algorithm in precision. Indeed Base Algorithm shows an overall average precision of 64.39, while enhanced algorithm obtains an average of 67.78. A t-test on the average results obtained by the two algorithms file by file through the ten simulations reveals how these differences are statistically significant (t = -6.719, p < 0.001). Moreover, enhanced algorithm should also be preferred to base algorithm because of its lower standard deviation, which guarantees more coherent results through the simulations. Results also show how there is a high variance in the results among the 19 files. The maximum difference between the two approaches has been found in file br-g18 where the precision obtained by enhanced algorithm was 7.74% above the precision obtained by base algorithm. In all runs enhanced algorithm outperforms base algorithm. Finally, the maximum precision values, obtained with file br-e27, show how enhanced algorithm is able to exceed the threshold of 80% (81.10%), while base algorithm is less precise with a 77.41%.

TABLE III. COMPARISON STATISTICS BETWEEN BASE AND ENHANCED ALGORITHMS

Algorithm	precision (mean)	precision (standard deviation)	maximum precision value
Base	64.39	6.1	77.41
Enhanced	67.78	5.8	81.10

It is interesting to note how there does not exist a relation between the number of nouns in a single sentence that has been analyzed and the corresponding precision results. Unexpectedly, base algorithm does not show correlation between the two variables (r = 0.120, p < 0.001), thus implying that context window size is not correlated to precision. An even more significant result is obtained with enhanced algorithm (r = 0.074, p < 0.001), supporting our initial experimental decision of running analyses sentence by sentence rather than paragraph by paragraph.

Despite the promising results, we noted that similarity alone is not sufficient. Indeed, usually there can be different possible set of senses of the nouns that are fairly plausible, in a given sentence.

TABLE IV. SIMILARITY DIFFERENCES BETWEEN SEMCOR SENSES AND BASE ALGORITHM'S BEST SOLUTION

	SemCor	Base Algorithm
Target	5	3
Chart	1	2
Additives	1	1
Similarity score	0.24	1.11

Table IV shows this drawback using the sentence "The target chart quickly and briefly tells you which additives do what." extracted from the file named br-e27 of the SemCor Corpus. By computing the overall similarity measure on both sets of senses, we obtain a value of 0.24 in the SemCor sense-tagged set and a value of 1.11 in the set computed by our base algorithm. The meaning of the senses in the latter set clearly indicates that words are strongly related: the fifth sense of "target" is "the goal intended to be attained" and the first sense of "chart" is "a visual display of information", while the third sense of "target" is "the location of the target that is to be hit" and the second sense of "chart" is "a map designed to assist navigation by air or sea". We are currently studying if and how this observation can be exploited in order to improve the precision of the disambiguation process.

Checking some results by hand we have also noted that in SemCor some words have been sense-tagged with a meaning that tough it is not totally wrong, it is at least ambiguous. This fact can be explained by an example. The file br-l18 contains the sentence "I asked her why she couldn't do it tomorrow, but it seems the muse is working good tonight and she's afraid to let it go" where the word "muse" has been tagged with the meaning of "in ancient Greek mythology any of 9 daughters of Zeus and Mnemosyne; protector of an art or science". Although this meaning is not completely wrong, it is definitely more correct the meaning of "the source of an artist's inspiration".

More generally, there are some cases where the sense assigned in SemCor is right, but, nevertheless not necessarily unambiguous. This observation raises the question whether the fine granularity of WordNet is appropriate for the word sense disambiguation task as discussed in [19].

V. RELATED WORK

The idea of using similarity among synsets in WordNet is not original. Much literature has tried to exploit WordNet semantic relations for WSD. In particular Zhang et al. [10] have implemented a genetic algorithm for noun disambiguation based on the Wu-Palmer measure of similarity and on SemCor word frequency. Their results are considerable as they obtained an overall 71.98% precision on the general SemCor testbase. Yarowsky [14] presented an unsupervised learning algorithm whose performance (overall 96% accuracy) is comparable to that of supervised algorithms. Yarowsky's algorithm applies two constraints to the properties of human language to discriminate among senses, i.e., one sense per collocation and one sense per discourse. Recently, Social Network Analysis has gained interest in WSD through the use of its classical graph connectivity metrics. Navigli and Lapata [2] used local centrality and global graph measures, showing that the former outperforms the latter and is comparable to the current state of the art. Unsupervised graph-based methods have been exploited also by Mihalcea [15]. In this work, synstet similarity is defined, similarly to Lesk [27], as a function of the number of common tokens in the definitions of word senses. This algorithm obtains an overall precision of 54.2%, being able to disambiguate nouns, and also verbs, adjectives, and adverbs. Recently, Navigli and Velardi [17] introduced the Structural Semantic Interconnections (SSI) algorithm that detects relevant semantic patterns of word senses through the use of a context-free grammar, obtaining a precision of 86% for nouns and almost 70% for verbs.

Several works have also attempted to use other resources in addition to WordNet [12]. In particular, they have focused on ontologies such as OntoNotes [18] and SUMO [16]. More specifically, these works are built on top of SVM-based supervised algorithms. Zhong et al. [19][20], based on OntoNotes, perform domain adaptation experiments trained using the knowledge sources of local collocations, part-of-speech, and surrounding words. The results of these papers highlight the importance of having an appropriate level of sense granularity. On the other hand, authors in [21] performed semantic disambiguation for Spanish. They used semantic classes instead of senses, based on the SUMO ontology. This approach allows collecting a larger number of examples for each class while polysemy is reduced, improving the accuracy of semantic disambiguation. In turn, works in [25][26] have proposed ways to exploit additional knowledge given by domain information. Specifically, Magnini et al. [26] proposed an extended version of WordNet called WordNet Domains obtaining an average 70% precision, while the work in [25] proposes a preliminary algorithm including domain information.

The need of an augmented version of WordNet has been formalized in [22] and [23]. The inclusion of logics and the exploitation of glosses to connect verbs and nouns have been explicitly called for. Naskar and Bandyopadhyay [24] have implemented a variation of the Lesk algorithm using eXtended WordNet [23] and its glosses to disambiguate nouns, verbs, and adjectives obtaining an 85% overall precision.

The main difference between our algorithm and other algorithms is that we are now able to deal with one of the main drawbacks of WordNet when using similarity measures, i.e., with the fact that the organization of words in hierarchies does not cross part of speech boundaries. Indeed, by extracting disambiguated nouns from the disambiguated glosses of verbs we create a new relation in WordNet that links each sense of each verb to one or more nouns, making it possible to process verbs through related nouns. This new kind of relation gives suggestions, in an automatic manner, about which nouns are used with which verbs in natural language.

VI. Conclusion and future work

In this paper, we presented a new algorithm that uses WordNet disambiguated glosses to create a relation between nouns and verbs in WordNet network. Our results suggest that the information provided by the new relation can be significantly helpful in the context of WSD. We have tested our algorithm on 19 randomly chosen SemCor files and we have found that it is able to outperform an algorithm based only on nouns and on Wu-Palmer similarity measure. Our algorithm has been able to reach an 81.10% precision on file br-e27 and an average of 67.78%.

The analysis of the results of our experiments also highlighted three main drawbacks: *(i)* though manually tagged, SemCor disambiguated words can present very ambiguous synsets that could even be considered wrong; *(ii)* as previous literature has pointed to, WordNet is a too fine-grained resource for WSD; *(iii)* the well-established methodology based on similarity often leads to wrong solutions since the right synsets are not necessarily the most similar. The first two mentioned issues strictly depend on the used tools, indeed, as shown in [19] precision would benefit from having a more coarse-grained resource such as OntoNotes [18]. Regarding the third issue, we are working in two directions: *(i)* on the development of a tool that allows the addition of contextual information to WordNet creating new types of relations, e.g., adjective-noun, further improving the presented Enhanced Algorithm; *(ii)* integrating the tool with domain-specific ontologies that could be used when dealing with documents in a specific context.

As a further development, we are considering to exploit domain knowledge. We have also run a preliminary version of new algorithms that include WordNet Domains in the WSD process, but they need to be refined since results are not promising, probably due to the nature of WordNet Domains which is too coarse-grained. Another important evolution of our algorithm is to focus on the disambiguation of other parts of speech, especially verbs, that could significantly help improve the overall sentence disambiguation, and adjectives, which could be useful for our application context, i.e., sentiment analysis.

References

[1] A. Budanitsky, and G. Hirst, “Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures,” In Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics. Pittsburgh, June 2001.

[2] R. Navigli, and M. Lapata, “An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation,” *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 4, pp. 678-692, Apr. 2010.

[3] J. Jiang, and D. Conrath, “Semantic similarity based on corpus statistics and lexical taxonomy,” Proc. International Conference on Research in Computational Linguistics. Taiwan, 1997.

[4] P. Resnik, “Using information content to evaluate semantic similarity in a taxonomy,” Proc. 14th International Joint Conference on Artificial Intelligence. Montreal, 1995.

[5] P. Resnik, and D. Yarowsky, “Distinguishing systems and distinguishing senses: new evaluation methods for Word Sense Disambiguation,” *Nat. Lang. Eng.* 5, 2, pp. 113-133, June 1999.

[6] T.H. Ng, “Getting Serious about Word Sense Disambiguation,” Proc. ACL SIGLEX Workshop Tagging Text with Lexical Semantics: Why, What, and How?, pp. 1-7, 1997

[7] D. Lin, “Using syntactic dependency as a local context to resolve word sense ambiguity,” Proc. 35th Annual Meeting of the Association for Computational Linguistics, pp. 64-71. Madrid, July 1997.

[8] Z. Wu, and M. Palmer, “Verb semantics and lexical selection,” Proc. 35th Annual Meeting of the Association for Computational Linguistics, pp. 133-138. Las Cruces, NM, July 1997.

[9] S. Banerjee, and T. Pedersen, “Extended gloss overlaps as a measure of semantic relatedness,” Proc. International Joint Conference on Artificial Intelligence, volume 18, pp. 805-810, 2003.

[10] C. Zhang, Y. Zhou, and T. Martin, “Genetic word sense disambiguation algorithm,” Proc. Second International Symposium on Intelligent Information Technology Application (IITA 08), 2008.

[11] S. N. Sivanandam, and S. N. Deepa, “Introduction to Genetic Algorithms,” Springer, 2008.

[12] C. Fellbaum, “WordNet: An electronic lexical database,” MIT Press, Cambridge, MA.

[13] Princeton WordNet Gloss Corpus. http://wordnet.princeton.edu/glosstag.shtml, last access: 07-19-2010.

[14] D. Yarowsky, “Unsupervised word sense disambiguation rivaling supervised methods,” In *Proceedings of the 33rd Annual Meeting on Association For Computational Linguistics*. Annual Meeting of the ACL. Association for Computational Linguistics, Morristown, NJ, pp. 189-196, June 1995.

[15] R. Mihalcea, “Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling,” In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Human Language Technology Conference. Association for Computational Linguistics, Morristown, NJ, pp. 411-418, 2005.

[16] I. Niles and A. Pease, “Towards a standard upper ontology,” In *Proceedings of the international Conference on Formal ontology in information Systems - Volume 2001*. FOIS '01. ACM, New York, NY, pp. 2-9, 2001.

[17] R. Navigli, and P. Velardi, “Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation,” *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 7, pp. 1075-1086, July 2005.

[18] S. S. Pradhan, E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, L., and R. Weischedel, “OntoNotes: A Unified Relational Semantic Representation,” In *Proceedings of the international Conference on Semantic Computing*. ICSC. IEEE Computer Society, Washington, DC, pp. 517-526, 2007.

[19] Z. Zhong, H. T. Ng, and Y. S. Chan, “Word sense disambiguation using OntoNotes: an empirical study,” In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Annual Meeting of the ACL. Association for Computational Linguistics, Morristown, NJ, pp. 1002-1010, 2008.

[20] Z. Zhong, and H. T. Ng, “Word sense disambiguation for all words without hard labor,” In *Proceedings of the 21st international Jont Conference on Artifical intelligence*. H. Kitano, Ed. International Joint Conference On Artificial Intelligence. Morgan Kaufmann Publishers, San Francisco, CA, pp. 1616-1621, 2009.

[21] R. Izquierdo-Bevia, L. Moreno-Monteagudo, B. Navarro, A. Suarez, “Spanish All-Words Semantic Class Disambiguation Using Cast3LB Corpus,” Lecture Notes in Computer Science numb 4293, Springer-Verlag, pp. 879-888, 2006.

[22] P. Clark, C. Fellbaum, J. R. Hobbs, P. Harrison, W. R. Murray, and J. Thompson, “Augmenting WordNet for deep understanding of text,” In *Proceedings of the 2008 Conference on Semantics in Text Processing*, ACL Workshops. Association for Computational Linguistics, Morristown, NJ, pp. 45-57, 2008.

[23] D. Moldovan, and V. Rus, “Explaining answers with extended wordnet,” In Proc. of ACL’01, 2001.

[24] S. K. Naskar, and S. Bandyopadhyay, “Word Sense Disambiguation Using Extended WordNet,” In *Proceedings of the international Conference on Computing: theory and Applications*, ICCTA. IEEE Computer Society, Washington, DC, pp. 446-450, 2007.

[25] S. G. Kolte, S. G. Bhirud, “Word Sense Disambiguation Using WordNet Domains,” In *Proceedings of the 2008 First international Conference on Emerging Trends in Engineering and Technology* ICETET. IEEE Computer Society, Washington, DC, pp. 1187-1191, 2008.

[26] B. Magnini, C. Strapparava, G. Pezzulo, and A. Gliozzo, “The role of domain information in Word Sense Disambiguation,” *Nat. Lang. Eng.* 8, 4, pp. 359-373, 2002.

[27] M. Lesk, “Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone,” In *Proceedings of the 5th Annual international Conference on Systems Documentation*. V. DeBuys, Ed. SIGDOC '86. ACM, New York, NY, pp. 24-26, 1986.

[28] Y. Karov, and S. Edelman, “Similarity-based word sense disambiguation,” *Comput. Linguist.* 24, 1, pp. 41-59, Mar. 1998.

[29] K, Meffert, et al., “JGAP – Java Genetic Algorithms and Genetic Programming Package”. http://jgap.sf.net.

[30] R. K. Belew, and M. D. Vose, “Foundations of genetic algorithms 4,” Morgan Kaufmann, 1997.

[31] D. Barbagallo, C. Cappiello, C. Francalanci, and M. Matera, “A Reputation-based DSS: the INTEREST Approach,” In *Proceedings of ENTER 2010: International Conference On Information Technology and Travel&Tourism*, February 2010.

[32] E. Agirre, and G. Rigau, "Word sense disambiguation using conceptual density," Proc. of COLING, 1996.

Categorisation of Semantic Web Applications

Michael Martin
University of Leipzig
Dept. Business Information Systems
Leipzig, Germany
martin@informatik.uni-leipzig.de

Sören Auer
University of Leipzig
Dept. Business Information Systems
Leipzig, Germany
auer@informatik.uni-leipzig.de

***Abstract*—The recent success of the Semantic Web in research, technology and standardisation communities has also resulted in a large variety of different standards, technologies and tools. This diversity and heterogeneity goes along with an increasing complexity in assessing, evaluating, selecting and combining different approaches for the development of Semantic Web Applications (SWA). With this work we aim at lowering the entrance barrier for the development and engineering of Semantic Web Applications by presenting a classification of SWAs according to the dimensions semantic technology depth, information flow direction, richness of knowledge representation, semantic integration and user involvement. This categorisation helps to establish and consolidate the conceptualisation with regard to the engineering of SWAs and facilitates the comparability of different SWAs. With its requirements and benefits, the categorisation of SWAs can also serve as a guideline for practitioners looking into the application of semantic technologies within their use cases. We give an overview over popular SWAs and present, with Vakantieland and LinkedGeoData, two semantic web applications with regard to the categorisation in detail.**

***Keywords*-Categorisation; Semantic Web; Web Applications;**

I. Introduction

Recently, we observed the Semantic Web and related technologies gaining traction. Oracle, for example, integrated support for semantic knowledge management into their database product [1], Google started to evaluate annotations [2] using Resource Description Framework attributes (RDFa) and the W3C has lately launched the second revision of the Web Ontology Language (OWL) standard [3].

The success of the Semantic Web in research, technology and standardisation communities has, however, also resulted in a large variety of different approaches, standards and techniques. For example, a variety of knowledge representation formalisms with different expressivity is available with RDF, RDF-Schema, and various OWL flavours; there exist different serialisations such as RDF/XML, N3, NTriple, RDFa, Trix; the semantic web technology space is complemented with a wealth of different reasoners, triples stores, rule processors, semantic web service infrastructures, various APIs, etc. This diversity and heterogeneity goes along with an increasing complexity in assessing, evaluating, selecting and combining different approaches. From a Web Engineering point of view, this diversity substantially enlarges the application space of semantic technologies, but at the same time complicates their application.

Compared to conventional Web Applications, Semantic Web Applications (SWA) employ a number of additional standards and technologies on the persistence, data interchange / transaction processing and user interface layers (cf. Table I). This work is based on defining a Web Application as a client-server software application, which uses the HTTP protocol for communication between client and server as well as user interface technologies, which common Web browsers are capable to process (i.e., often HTML, CSS and Javascript or to a lesser extend UI technologies such as SVG or proprietary equivalents such as Flash and Silverlight). Our definition of a Semantic Web Application extends the Web Application definition with the requirement of using some Semantic Web knowledge representation formalism at either one or multiple of the persistence, data interchange / transaction processing and user interface layers. Semantic Web knowledge representation formalisms are mostly based on the RDF data model and include standards such as RDF-Schema, OWL, RIF or RDFa. The use of semantic technologies has a great potential in particular for the adaptability of Web applications, the efficient and standardized syndication of structured information or for improved search within and across different SWAs.

With this work we aim at *lowering the entrance barrier* for the development and engineering of SWAs by presenting a classification of SWAs according to the dimensions semantic technology depth, information flow direction, richness of knowledge representation, semantic integration and user involvement. This categorisation helps to establish and consolidate the conceptualisation with regard to the engineering of SWAs and facilitates the comparability of different SWAs. With the description of requirements and benefits for each of the different characteristics, the categorisation of SWAs can also serve as a guideline for practitioners looking into the application of semantic technologies within their use cases.

The paper is structured as follows: We describe our categorisation model along a number of dimensions in Section II. We present an overview of popular Semantic Web Applications in the light of these categorisations together with an in-depth description of two particular Semantic Web

Table I
JUXTAPOSITION OF CONVENTIONAL AND SEMANTIC WEB APPLICATION TECHNOLOGIES.

	Web Application	**Semantic Web Application**
Persistence Layer	Relational Database, ODBC,SQL	Triple Store, ODBC, SPARQL
Data Interchange & Transaction processing	REST-APIs, Web Services	SPARQL & LinkedData endpoint, Semantic Web Services
User Interface	(X)HTML, CSS, JS	(X)HTML, CSS, JS, RDFa, GRDDL

applications in Section III. We conclude and present related as well as future work in the Sections IV and V.

II. CATEGORISATION OF SEMANTIC WEB APPLICATIONS

In this section we discuss a number of dimensions along which semantic web applications can be characterised. These dimensions are the depth of the application architecture to which semantic technologies are applied, the direction(s) of semantic information flows, the richness of semantic knowledge representations, the intensity of the semantic integration with other SWAs and representation formalisms as well as the degree of user involvement.

A. Semantic Technology Depth

This categorisation dimension aims to capture to which degree the architecture of an SWA makes use of semantic technologies. Generally, SWAs can use semantic technologies in two different ways – externally and/or internally:

Extrinsic SWA: make use of semantic knowledge representation formalisms on the surface of the application in order to facilitate the interaction and integration with other SWAs and technologies. Implementation-wise, extrinsic SWAs are easy to realise, since conventional Web application development technologies and design patterns can be used. In order to map between internal persistence data models and semantic web taxonomies, vocabularies and ontologies, a number of tools exist [4]. Of particular importance are relational database schema, since their use is widespread, not only with Web applications. A comprehensive overview on approaches and technologies for transforming relational data to RDF is contained in [5]. Recently, the Linked Data paradigm has attracted quite some attention for exchanging and integrating data over the Web. Based on a relational to RDF mapping, Web applications can be easily equipped with a linked data interface (cf. e.g., [6]). Another popular approach to equip Web applications with a Semantic Web interface is RDFa standard [7] (sometimes also subsumed under Linked Data), which defines how conventional HTML can be annotated with RDF.

Intrinsic SWA: make direct internal use of semantic representations for their original application architecture. Here the situation is more complicated than with solely extrinsic SWA, since conventional technologies have to be complemented or replaced by their Semantic Web equivalents. On the persistence layer relational databases have to be replaced by triple stores. On the API layer Object-Relational-Mapping (ORM) techniques have to be replaced by corresponding APIs, which provide higher-level functions for handling RDF, RDF-Schema and OWL. In particular RDF data management, i.e., the querying performance of triple stores, is a decisive factor for the intrinsic use of semantic technologies in SWA (cf. e.g., [8], [9]). In recent years much progress has been made to improve the performance of triple stores by developing better storage, indexing and query optimisation. However, compared to querying data stored in a fixed relational database schema, querying a triple store is still usually slower by a factor of 5-50 (cf. e.g., BSBM results [10]). This shortcoming is due to the fact that columns in a relational database are typed and may be indexed more efficiently. By using a triple store, this efficiency is lost to the flexibility of amending and reorganising schema structures easily and quickly.

B. Information Flow Direction

The class of extrinsic SWAs can be further refined into SWAs, which produce, consume or produce and consume semantic representations.

Producing SWA: Based on either an intrinsic semantic information representation or on a mapping of other data models to RDF (as discussed in the previous section), four different types of Semantic Web interfaces can be distinguished:

- ETL-style dumping of information in RDF,
- provisioning of Linked Data, RDFa or GRDDL interfaces,
- declarative querying e.g., by means of SPARQL endpoints,
- Semantic Web Services or REST-style APIs, which return structured information adhering to the RDF data model.

The provisioning of semantically represented information in one of these forms helps to distribute and syndicate structured content. In particular, the re-usability and re-purposability of information is facilitated. Compared to REST APIs and Web Services returning information in proprietary formats, these interfaces provide standardized means for accessing structured information. In order to build mashups, which combine information for various sources, Web developers would (when enabled to use one of these SWA interfaces) not be required to get acquainted with with various APIs and result formats. However, only REST APIs and Web Services are suited for transaction processing.

Consuming SWA: Information published as RDF is reusable by SWAs. If an SWA accesses information from the Data Web to enrich there own information space, it is classified as a Consuming SWA. A Consuming SWA can obtain information from either one or multiple of the methods used for publishing structured information used by producing SWAs. In most cases it will be sufficient for a consuming SWA to retrieve information via the HTTP protocol and parse one or multiple of the result formats RDF serializations, RDFa or SPARQL result formats. If producing SWAs offer RDF serialized according to the JSON specification [11], even specific parsing is not required, since JSON parsers are part of the standard functionality of most programming languages.

C. Richness of Knowledge Representation

SWAs can be further classified according to their use of rich knowledge representation formalisms:

- *Shallow KR SWA.* Comprise SWA, which e.g., primarily use taxonomies, simple hierarchies and relatively simple knowledge representation formalisms such as RDF and RDF-Schema.
- *Strong KR SWA.* Comprise SWA, which use higher level knowledge representation formalism such as different OWL variants, rules etc.

A navigator for the expressivity and complexity of description logics is also available [12]. Already the declarative querying of knowledge bases by means of SPARQL currently adds a substantial performance overhead to SWAs compared to relational database backed Web applications without even considering implicit information, which is must be revealed by reasoning. This is why we do not expect comprehensive description logic reasoning to be part of standard SWAs in the short to medium term. Instead there might be some light inferencing, which can be performed (on demand or in certain intervalls) by executing inference rules directly within triple stores (e.g., for resolving co-references, inverse relationships and computing transitive closures).

D. Semantic Integration

This categorisation dimension measures how well an SWA is integrated within the Semantic Web. The integration can be measured on the schema and instance level. On the schema level, for example, the number of overall schema elements (i.e., RDF/OWL classes and properties) can be put in relation to the number of reused schema elements, i.e., schema elements, which are either defined elsewhere or for which a `owl:sameAs` relation with an external element is defined.

Similarly, we can measure the semantic integration on the instance level. Semantic integration on the schema level appears to be slightly more important, than instance level integration, since in most cases there are more SWA, which publish information of a certain type (e.g., about Cities), than SWA, which publish information about a certain entity (e.g., Vienna).

For the integration and reuse on the schema level the availability of suitable upper level ontologies is important. For the semantic integration on the instance level interlinking hubs or crystallization ontologies such as DBpedia [13] are crucial. Depending on the level of semantic integration, we call representatives integrated (respectively isolated):

- *Isolated SWA* are categorized by a limited reuse of shared identifiers, vocabularies and ontologies.
- *Integrated SWA* are categorized by a strong reuse of shared identifiers, vocabularies and ontologies.

E. User Involvement

Another important characteristic of SWAs is the degree of end-user involvement. End-users can be roughly classified into spontaneous contributors, advanced users and knowledge engineering experts. Subsequently, an SWA can be categorized according to the sizes and ratios in which these different end-user groups are participating in the creation of semantic knowledge representations within an SWA. Also, it can be made clear which of these groups are restricted to contributions on the instance level and which participate in refining the knowledge schema. Other facets of the user involvement, which are not specific to SWAs are for example: the degree of closed user group, free for all, edit functionality for all information or just parts of the content.

F. Requirements and Benefits of characterization dimensions

We give an overview of the requirements and benefits of the presented categorisation dimensions for the implementation of SWAs in Table II. Based on the categorization dimensions different classes of SWAs can be distinguished:

- **Search engine / crawler.** Semantic search engines / crawler are extrinsic SWAs with a consuming information flow direction and mostly a shallow semantic richness. If such SWAs also process and republish retrieved RDF information, they can be considered as semantically integrated.
- **Collaborative knowledge acquisition.** Representatives of this class of SWAs are usually tailored towards a certain knowledge domain, although generic applications such as Semantic Wikis falling into that category exist. SWAs in this class are community oriented, mostly extrinsic and intrinsic, have a producing information flow direction and are often semantically integrated.
- **Visualization oriented**: SWAs of this visualization oriented class heavily use own or extrinsic retrieval and publish the received information with regards to a certain usage scenario and environment. Such SWAs have a consuming information flow direction and are semantically integrated.

Table II
SWA CHARACTERISATION OVERVIEW INDICATING REQUIREMENTS AND BENEFITS.

Dimension	Requirements	Benefits
Semantic technology depth		
Extrinsic	mapping between internal information structures and RDF	standardised interaction
Intrinsic	sufficient query processing power	increased schema flexibility
Information flow direction		
Consuming	mapping of RDF to internal information structures	wealth of additional structured information
Producing	mapping of internal information structures to RDF	increased information distribution
Semantic richness		
Shallow	availability of structured information	pay-as-you go strategy
Strong	comprehensive knowledge engineering	automated reasoning
Semantic integration		
Isolated	creation of own vocabularies and ontologies	simplified information governance
Integrated	vocabulary and identifier reuse on schema and/or instance level, co-reference and matching techniques	simplified syndication of semantic content
User involvement		
Com.-oriented	provisioning of simple interaction with semantic content	exploitation of crowd intelligence

- **Information Chaining**: SWAs of this class give users the possibility to get connected information from different distributed information spaces. In this case SWAs are extrinsic and have a consuming as well as a producing information flow direction. Furthermore, they are also intrinsic and semantically integrated, because they mostly store and process the received information.

III. Categorisation Examples

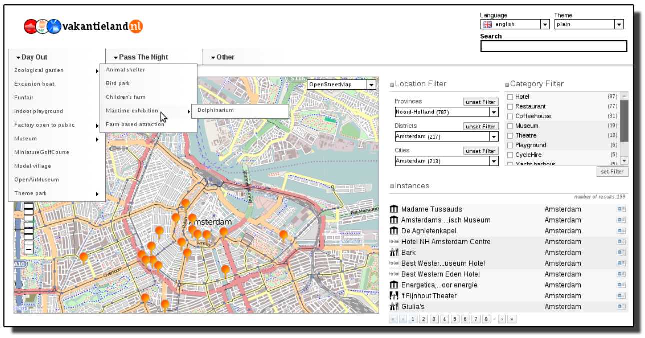
In this section we present an overview of existing SWAs according to the categorisation dimensions.

The selected SWAs are representatives of the exising SWA landscape, whose categorisations are presented in Table III. Some of the presented SWAs in this table, such as OntoWiki and Semantic Media Wiki, can not be categorised unambiguously. These SWAs are used to handle information of different domains in ways that the used vocabularies are defined elsewhere or created for the first time. However, if instances of such SWAs (i.e., the OntoWiki of the Leipzig Professors Catalogue [14] or OpenResearch [15] based on Semantic Media Wiki) will be investigated, it is possible to determine the correct classifications for the specific categorisation dimensions.

In the following we present two SWAs in more detail in order to explain the categorisation dimensions at an example.

A. Vakantieland

Vakantieland [16] publishes comprehensive information about 20,000 touristic points-of-interest (POI) in the Netherlands such as textual descriptions, location information and tourism features. The information is stored in a knowledge base containing almost 2 million triples. The Vakantieland data is structured using approximately 1,250 properties as well as 400 classes, which are used among others to provide different search and filter functionalities. As illustrated in Figure 1 it is possible to select a set of tourism classes which can be combined with other filter criteria such as terms from the free-text search as well as elements of the spatial hierarchy.



Madame Tussauds	Amsterdam
Amsterdams ...isch Museum	Amsterdam
De Agnietenkapel	Amsterdam
Hotel NH Amsterdam Centre	Amsterdam
Bark	Amsterdam
Best Wester...useum Hotel	Amsterdam
Best Western Eden Hotel	Amsterdam
Energetica,...oor energie	Amsterdam
't Fijnhout Theater	Amsterdam
Giulia's	Amsterdam

Figure 1. The Vakantieland Semantic Web Application.

The depicted map acts also as an interactive map-bounding-box filter. According to the search and filter criteria a set of POIs is then being presented. Every POI description of such a result set can also be visited on a separate details page, consisting of properties arranged in a property hierarchy.

- **Semantic technology depth:** Vakantieland is an intrinsic *and* extrinsic SWA, since it employs the RDF data model for internal representation of information. Its implementation is based on the *Erfurt* Semantic Web API. With regard to publication, Vakantieland provides a Linked Data interface, which includes RDFa.
- **Information flow direction:** The POIs presented in Vakantieland were stored formerly in a relational database. While redesigning this application as an SWA, the data was converted to RDF and stored

Table III
EXAMPLES OF SWAS CATEGORISED ALONG THE CATEGORISATION DIMENSIONS.

Application	**STD Extrinsic**	**STD Intrinsic**	**IFD Consuming**	**IFD Producing**	**SR Shallow**	**SR Strong**	**SI Isolated**	**SI Integrated**	**UI Com. Oriented**
Collaborative Knowledge Aquisition									
OntoWiki (http://www.ontowiki.net)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Revyu (http://www.revyu.com/)	✓	✓	-	✓	✓	-	-	✓	✓
Semantic Media Wiki (http://www.semantic-mediawiki.org/)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Information Chaining									
Deri Pipes (http://pipes.deri.org/)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Freebase (http://www.freebase.com/)	✓	✓	✓	✓	✓	-	-	✓	✓
Twine (http://www.twine.com/)	✓	✓	-	✓	-	✓	-	✓	✓
Search Engines									
Bing reference search (http://www.bing.com/reference)	✓	✓	✓	-	✓	-	-	✓	-
Geonames (http://www.geonames.org/)	✓	✓	n/a	✓	✓	n/a	-	✓	-
Google Squarred (http://www.google.com/squared)	✓	n/a	✓	-	✓	-	-	✓	-
Sig.ma (http://sig.ma/)	✓	✓	✓	-	✓	-	-	✓	-
Sindice (http://www.sindice.org/)	✓	✓	✓	-	✓	-	-	✓	-
Swotti (http://www.swotti.com/)	✓	n/a	✓	-	n/a	n/a	-	✓	✓
Visualization Oriented									
DBpedia Mobile (http://wiki.dbpedia.org/DBpediaMobile)	✓	-	✓	-	-	✓	-	✓	-
Facetted Wikipedia Search (http://dbpedia.neofonie.de/browse/)	✓	-	✓	-	-	✓	-	✓	-
RelFinder (http://relfinder.dbpedia.org/)	✓	-	✓	-	-	✓	-	✓	-

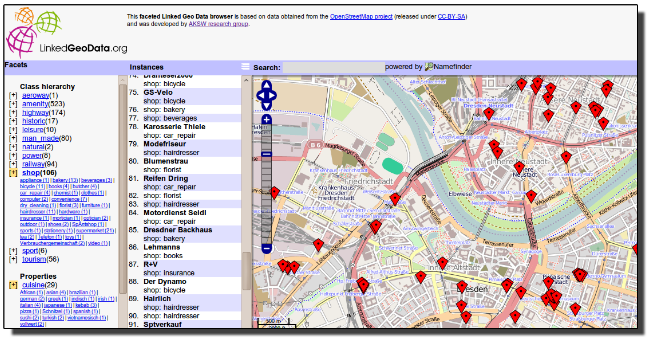
in a Triple-Store (OpenLink Virtuoso). In addition to publish the information for end-users with HTML/CSS/JS, the information is also provided as RDF (RDF/XML, Turtle, N3, JSON), which demonstrates that Vakantieland is a *producing* SWA. Except the geo-coordinates, which are retrieved from different geocoding services, Vakantieland does not consume RDF data from other SPARQL or LinkedData endpoints at this time.

- **Semantic richness:** With regard to the expressivity of the used knowledge representation techniques, Vakantieland is rather constrained and mostly in the RDF and RDF-Schema space. The used OWL features are confined to class and property definitions. In this case the semantic richness of this information space can be categorised as *shallow*.
- **Semantic integration:** The semantic integration is medium. On the schema level Vakantieland reuses vocabularies such as DublinCore [17], WGS84 [18] and GoodRelations [19], but also defines a large number of own schema elements, such as tourism classes, tourism object features, tourism offerings as well as different address, geospatial and contact properties. In future Vakantieland will become a fully integrated SWA since it is planned to link instances with DBpedia resources.
- **User involvement:** Vakantieland is a moderated tourism Wiki. At the moment, it is possible to edit fulltext-descriptions, address and contact information, which already helps to decrease costs for maintaining and to increase the quality of presented information. Only predefinied properties are editable by end users. An appropriate moderation process will be included to prevent publication of inappropriate material. Vakantieland is *community oriented* but not as much as other Semantic Wikis.

B. LinkedGeoData

LinkedGeoData is an effort to add a spatial dimension to the Web of Data. LinkedGeoData uses the information collected by the OpenStreetMap project and makes it available as an RDF knowledge base according to the Linked Data principles. It interlinks this data with other knowledge bases in the Linking Open Data initiative. The benefits of revealing the structured information in OSM are accessible in a faceted based browser [20] as depicted in Figure 2.

This user interface allows to browse the world by using a slippy map. Once a region is selected, the browser analyses the descriptions of nodes and ways in that region and generates facets for filtering. Once a facet or a specific facet value has been selected, matching elements are displayed as markers on the map and in a list. If the selected region is changed, these are updated accordingly. If a user logs into the application by using her OSM credentials, the displayed elements can directly be edited in the map view. For this, the browser generates a dynamic form based on existing properties. The form also allows to add arbitrary additional



This **faceted Linked Geo Data browser** is based on data obtained from the OpenStreetMap project (released under CC-BY-SA) and was developed by AKSW research group.

Class hierarchy
[+] aeroway(1)
[+] amenity(523)
[+] highway(174)
[+] historic(17)
[+] leisure(10)
[+] man_made(80)
[+] natural(2)
[+] power(8)
[+] railway(94)
[+] **shop(106)**
appliance (1) | bakery (13) | beverages (3) | bicycle (11) | books (4) | butcher (4) | car_repair (4) | chemist (1) | clothes (1) | computer (2) | convenience (7) | dry_cleaning (1) | florist (3) | furniture (1) | hairdresser (11) | hardware (1) | insurance (1) | mortician (1) | optician (2) | outdoor (1) | shoes (2) | SpÃ¤tshop (1) | sports (1) | stationery (1) | supermarket (21) | tea (2) | Telefon (1) | toys (1) | Verbrauchergemeinschaft (2) | video (1) |
[+] sport(6)
[+] tourism(56)

Properties
[+] cuisine(29)
African (1) | asian (4) | brazilian (1) | german (2) | greek (1) | indisch (1) | irish (1) | italian (4) | japanese (1) | kebab (3) | pizza (1) | Schnitzel (1) | spanish (1) | sushi (2) | turkish (2) | vietnamesisch (1) | vollwert (2) |

74. **Drahtesel2000**
shop: bicycle
75. **GS-Velo**
shop: bicycle
76. shop: bakery
77. shop: beverages
78. **Karosserie Thiele**
shop: car_repair
79. **Modefriseur**
shop: hairdresser
80. **Blumenstrau**
shop: florist
81. **Reifen Dring**
shop: car_repair
82. shop: florist
83. shop: hairdresser
84. **Motordienst Seidl**
shop: car_repair
85. **Dresdner Backhaus**
shop: bakery
86. **Lehmanns**
shop: books
87. **R+V**
shop: insurance
88. **Der Dynamo**
shop: bicycle
89. **Hairlich**
shop: hairdresser
90. shop: hairdresser
91. **Sptverkauf**

Figure 2. The LinkedGeoData browser Semantic Web Application.

properties. In order to encourage reuse of both properties and property values, the editor performs a type-ahead search for existing properties and property values and ranks them according to the usage frequency. When changes are made, these are stored locally and propagated to the main OSM database by using the OSM API.

- **Semantic technology depth:** The LinkedGeoData browser uses a a data model in its persistence layer, which is close to the RDF data model, but at the same time also more tailored towards the specific requirements (e.g., handling of large volumes of semantically annotated geospatial data). Hence, the LGD browser represents some hybrid type with regard to the semantic technology depth. Since the LGD browser also offers LinkedData and SPARQL interfaces it can, however, be characterized to be extrinsic.
- **Information flow direction:** The LGD browser is primarily a producing SWA. However, it also draws substantially from OpenStreetMaps data (which uses a relational representation).
- **Semantic richness:** The LGD knowledge bases use very shallow KR formalisms, mostly RDF and RDF-Schema. Ontology reasoning is not feasible regarding the size of LGD (with more than 3 billion triples).
- **Semantic integration:** The semantic integration of LGD is still rather low, since most of the data (e.g., streets, buildings, areas etc.) and schema elements (taxonomies of spatial objects and categorisations) in LGD are still relatively unique on the Data Web. However, LGD uses a few vocabulary elements (e.g., from the W3Cs WGS vocabulary) and is interlinked with DBpedia.
- **User involvement:** LGD itself has a relatively small and rather passive user community. However, it substantially draws from the vast OpenStreetMaps community, which is also the reason, why the KR formalisms are rather shallow.

IV. Related Work

Other than for the engineering and development of Web Applications (e.g., [21], [22], few approaches specifically tailored for the engineering Semantic Web applications exist. The *Semantic Web Framework* (SWF), for example, is a component-based framework for rapidly analysing required components, the dependencies between them, and selecting existing solutions [23]. A characterization of large scale semantic applications is presented in [24]. Based on this characterization, a guideline for the specification and design of large scale semantic applications was developed. Other than the work presented in this paper, the characterization and guidelines focus on large semantic applications in general and are not specifically tailored towards smaller SWAs. Another approach tackling the design and development of Semantic Web Application based on existing standards was published in [25]. This work represents a framework for engineering SWAs, that spans over several enterprises by applying techniques, methodologies, and notations offered by software engineering, Web engineering, and Business Process modelling. Existing Web Engineering processes are about design, implementation and maintenance of Web Applications, but lack the generation of meta-data. The "Web Engineering for Semantic Web Applications" (WEESA) approach [26] particularly tackles this aspect.

V. Conclusions and Future Work

While the applicability of semantic technologies was substantially broadened by the growth of Semantic Web standards, tools and approaches, the engineering complexity of SWAs substantially increased. With this work we aimed to contribute, to establish and to consolidate the conceptualisation of SWAs and facilitate the comparability of different SWAs. One of the intentions of using formal knowledge representation techniques (such as ontologies) is the decoupling of data and the application and the transition to flexible interfaces between both. However, a complete separation between information structures and application logic will not be completely possible. Hence, it is paramount to outline methodologies for the co-design of SWAs and knowledge bases. In the next paragraphs we outline the from our point of view most pressing hurdles for the wide-spread adoption of SWAs.

Closing the performance gap between relational and RDF data management: It has been widely acknowledged that the querying performance of triple stores is a decisive factor for the large-scale deployment of semantic technologies in many usage scenarios (cf. e.g., [8], [9]). In recent years much progress has been made to improve the performance of triple stores by developing better storage, indexing and query optimization. However, compared to querying data stored in a fixed relational database schema, querying a triple store is still usually slower by a factor of 2-20 (cf. e.g., BSBM results in [10]). This shortcoming is due to the fact that columns in a relational database are typed and may be indexed more efficiently. By using a triple store, this efficiency is lost to the flexibility of amending and reorganizing schema structures easily and quickly. A circumstance currently not yet taken advantage of by triple stores is that in typical application scenarios only relatively small parts of a knowledge base change within a short period of time. Based on this observation SPARQL result caching and view materialization strategies can be developed, which accelerate access to frequently used information structures.

Authoring of semantic-rich content: The overwhelming success of the World Wide Web was to a large extend based on the ability of ordinary users to author content easily. In order to publish content on the WWW, users had to do little more than to annotate text files with few, easy-to-learn HTML tags. Unfortunately, on the semantic data web the situation is slightly more complicated. Users do not only have to learn a new syntax (such as N3, RDF/XML or RDFa), but also have to get acquainted with the RDF data model, ontology languages (such as RDF-S, OWL) and a growing collection of connected RDF vocabularies for different use cases, such as FOAF, SKOS and SIOC. Previously, many approaches were developed to ease the syntax side of semantic authoring [27], [28]. In order to enable ordinary users to author rich semantic representations easily, user interfaces of SWAs have to also hide the data model from ordinary users without giving up the flexibility of mixing and mashing different, evolving vocabularies.

REFERENCES

[1] X. Lopez and S. Das, “Oracle database 11g semantic technologies, semantic data integration for the enterprise, white paper,” Oracle Semantic Technologies Center, Tech. Rep., 2009.

[2] “Introducing rich snippets,” http://googlewebmastercentral.blogspot.com/2009/05/introducing-rich-snippets.html.

[3] M. Schneider, “Owl 2 web ontology language rdf-based semantics,” W3C Rec., Tech. Rep., 2009.

[4] “Converter To Rdf,” http://esw.w3.org/topic/ConverterToRdf.

[5] S. S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. Thibodeau, S. Auer, J. Sequeda, and A. Ezzat, “A Survey of Current Approaches for Mapping of Relational Databases to RDF,” 2009.

[6] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumueller, “Triplify: light-weight linked data publication from relational databases.” in *WWW2009*. ACM, pp. 621–630.

[7] B. Adida, M. Birbeck, S. McCarron, and S. Pemberton, “RDFa in XHTML: Syntax and Processing,” W3C, Rec., 2008.

[8] M. Schmidt, T. Hornung, G. Lausen, and C. Pinkel, “Sp2bench: A sparql performance benchmark,” in *ICDE*. IEEE, 2009.

[9] C. Bizer and A. Schultz, “The Berlin SPARQL Benchmark,” *International Journal On Semantic Web and Information Systems*, 2009.

[10] “Berlin SPARQL Benchmark,” http://www4.wiwiss.fu-berlin.de/bizer/BerlinSPARQLBenchmark/.

[11] “RDF JSON Specification,” http://n2.talis.com/wiki/RDF_JSON_Specification.

[12] “Complexity of reasoning in Description Logics,” http://www.cs.man.ac.uk/~ezolin/dl/.

[13] J. Lehmann, C. Bizer, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, “DBpedia - a crystallization point for the web of data,” *Journal of Web Semantics*, vol. 7, no. 3, pp. 154–165, 2009.

[14] T. Riechert, U. Morgenstern, S. Auer, S. Tramp, and M. Martin, “Knowledge engineering for historians on the example of the catalogus professorum lipsiensis,” in *ISWC2010*, ser. LNCS. Shanghai / China: Springer, 2010.

[15] “Open Research,” http://www.openresearch.org/.

[16] “Vakantieland Prototypes,” http://staging.vakantieland.nl/.

[17] “The Dublin Core Metadata Initiative,” http://dublincore.org/.

[18] “Basic Geo (WGS84 lat/long) Vocabulary,” http://www.w3.org/2003/01/geo/.

[19] “GoodRelations - The Web Ontology for E-Commerce,” http://purl.org/goodrelations/.

[20] “LinkedGeoData Browser,” http://browser.linkedgeodata.org/.

[21] L. S. Al-Salem and A. Abu Samaha, “Eliciting web application requirements - an industrial case study,” *J. Syst. Softw.*, vol. 80, pp. 294–313, 2007.

[22] F. Daniel and M. Matera, “Turning web applications into mashup components: Issues, models, and solutions. ICWE2009,” ser. LNCS, vol. 5648. Springer, pp. 45–60.

[23] R. García-Castro, A. Gómez-Pérez, O. M. noz García, and L. Nixon, “Towards a component-based framework for developing semantic web applications,” 2008, pp. 197–211.

[24] O. M. noz García and R. García-Castro, “Guidelines for the specification and design of large-scale semantic applications. ASWC2009,” 2009, pp. 184–198.

[25] M. Brambilla, I. Celino, S. Ceri, D. Cerizza, E. D. Valle, and F. M. Facca, “A software engineering approach to design and development of semantic web service applications. ISWC2006,” ser. LNCS, vol. 4273, pp. 172–186.

[26] G. Reif, H. Gall, and M. Jazayeri, “Weesa: Web engineering for semantic web applications. WWW2005.” ACM, pp. 722–729.

[27] T. Tudorache, N. F. Noy, S. Tu, and M. A. Musen, “Supporting Collaborative Ontology Development in Protégé,” in *ISWC2008*, ser. LNCS, vol. 5318. Springer, 2008.

[28] S. Auer, S. Dietzold, and T. Riechert, “OntoWiki – A Tool for Social, Semantic Collaboration,” in *ISWC2006*, ser. LNCS, vol. 4273. Springer, 2006.

WebTribe: Implicit Community Clustering by Semantic Analysis

Damien Leprovost
Le2I - CNRS Lab.
University of Bourgogne
Dijon, France
damien.leprovost(at)u-bourgogne(dot)fr

***Abstract*—Since the advent of Web 2.0, any user becomes a content provider through personal websites, posts on wikis and forums, recommendations, annotations, etc. In this paper, we propose a method to analyze the interests of users based on their publishing activities, by positioning them into a semantic graph. We describe the WebTribe system that allows to extract topic information from collaborative websites and to query the resulting clusters of users.**

***Keywords*-implicit communities; semantic distance; topic graph.**

I. Introduction

The overflowing data produced by collaborative websites (forums, wikis, etc.) requires new analysis tools. Now, any Web user is no longer a simple reader, but a content provider who publishes information on the network: he is able to share his opinion. Such new data offers new opportunities, and must be analyzed.

In these circumstances, the indexing methods proposed by traditional systems such as user profiles ([1], [2]) may suffer limitations. Indeed, the description of a person's activities, whether by itself or by others, is often simplistic: users are reluctant to spend a precious time filling their profiles. User profiles do not define their precise interests, from the strongest to the more tenuous one, as manifested by the user's activities. Furthermore, profiles often static and can not be updated at any time. We therefore rely on an implicit definition of user interests to detect his/her activities properly.

Our goal in this paper is to identify implicit communities, that focuses on specific topics. Members of these communities are not necessarily aware of their membership, or even of the existence of the community. Indeed, what a user seeks is not necessarily in contact with him. In this sense, implicit communities are strongly apart from communities as they exist in social networks.

The paper is organized as follows: we present the architecture of our system in Section II. Section III defines the semantic topic graph that we construct. Section IV describes how the user is integrated into the graph and the graph querying possibilities. We present the system milestones in Section V. Section VI sums up the related work and we conclude in Section VII.

II. Architecture

We briefly present each analysis step of the WebTribe system, and will explicit them in following section. Figure 1 presents the flowchart for our proposal.

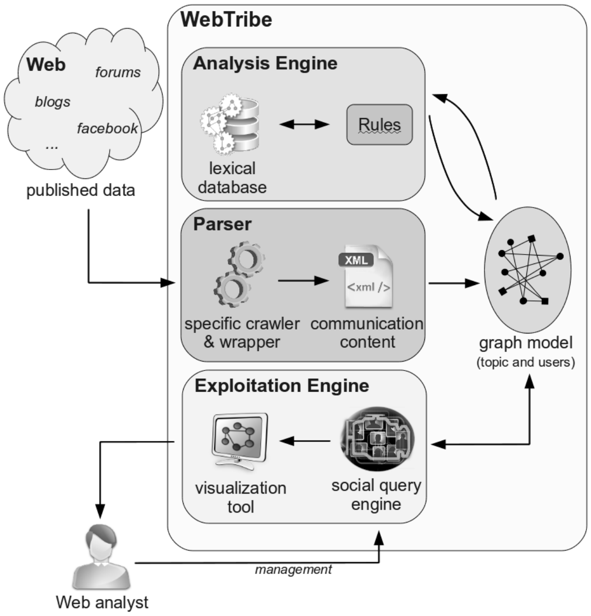


Figure 1. Architecture used for community clustering.

WebTribe has for input various published data on the web, and is managed by a Web analyst, who controls the system. WebTribe is structured around a graph model, and has three internal layers : the *Parser*, which extracts content from given sources; the *Analysis Engine*, which interprets the meaning of content; and the *Exploitation Engine*, which builds communities according to parameters and wishes of the Web analyst.

In the first step, a web analyst provides a list of topic used to build a topic graph, as the basis of our analysis. This list is the lexical database to be used by the *Analysis Engine* to find related content on the analyzed documents. This graph will be potentially pruned for non-relevant topics, and used for semantic user positioning.

In parallel, the *Parser* collects various publications (posts, etc.) from various sources selected by the web analyst, and

ISBN: 978-1-61208-104-5

associates the publication with its author.

Then, the *Analysis Engine* extracts for each publication its main topics, and quantifies the *publication attractivity* by topics, as the degree of importance of each topic evaluated in the publication. By analyzing all publication found for one author, we are now able to compute the *user attractivity* of this author. Using a Web-based semantic distance computing method (see Section IV), we evaluate the distance between topics and locate the user inside this topic graph.

Finally, querying the system through *Exploitation Engine* now means to compute a sub-graph of our results, including users who validate a closeness constraint given by the query, based on previous computed semantic distance.

The system is equipped with a query language and visualization tools that allow the Web analyst to explore sets of users.

III. Topic graph

A. Choosing topics

Our method aims to group users according to their affinities with defined topics. The Web analyst has to define which major topic are relevant for the analysis of his system. We call this topic list the *lexicon* of the system. The goal is to have enough topics to cover all of users. But having too many topics is not desirable either, unnecessarily burdening the system. We propose, at the end of this section, a method for pruning topics so that only useful topics remain.

Example 1: The Web analyst of a car fan forum submits the following lexicon: `Ferrari`, `Porsche`, `tuning`, `petrol`, `dealership`, `engine` and `fuel`.

B. Topic graph

Once defined all system topics, we have to organize them. To put them all into a weighted semantic graph, we use a Web-based semantic distance computing method [3], to evaluate the semantic distance between a term x and a term y. This method is well suited for our approach, because it does not extract the semantic distances from predefined ontologies, but from the Web content (through what Google sees, which seems to be the best viewpoint available). Since we intend to bring together users based on their activity on the Web, this method seems very appropriate to our context.

Therefore, the semantic distance between x and y is defined as follow:

$$\textsc{Dist}(x,y) = \frac{max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log M - min\{\log f(x), \log f(y)\}},$$

where $f(\lambda)$ is the frequency of the term, and M the total number of indexed terms. Using Google, $f(\lambda)$ means that the number of results to the "λ" query, and M the number of documents indexed (estimated at 1 trillion).

This expression calculates the lowest probability of $x|y$ and $y|x$, where $|$ means conditional probability, using a negative logarithm to increase the difference significance, and standardized by a division to solve scale problems.

Finally, our topic graph is a complete graph with topics as vertices. An edge between topics t_i and t_j is annotated by their distance.

Example 2: With previous lexicon, WebTribe computes the following semantic distances:

	fuel	engine	dealership
Ferrari	1,3478	1,6431	1,0418
Porsche	1,1140	1,4399	0,9475
tuning	1,3064	1,4529	0,7161
dealership	0,8998	1,1027	-
engine	1,0774	-	-

	tuning	porsche
Ferrari	1,3010	0,4195
Porsche	1,1301	-

Table I
Computed distances using examples' lexicon

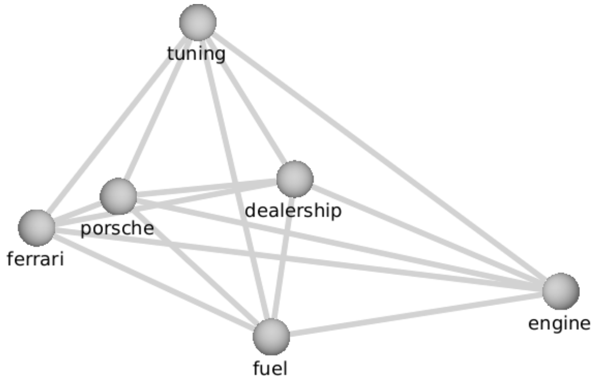


Figure 2. Example of Topic Graph

C. Pruning topics

The resulting graph of topics is a complete graph. In order to be as relevant as possible, but also be easily used, it must be as small as possible. Indeed, the more the number of topics is small compared to the number of content analyzed, the more shades of distances between users have an interesting meaning.

For these reasons, we prune topics considered non-relevant. A topic is not relevant when it is too close to another semantically. That is, when distance is smaller that a threshold δ_s given by the Web analyst. When this happens, the topic with the lowest frequency is removed. In other words, the remaining topic is considered to represent a concept encompassing the pruned topics. This reduces the graph size, making it more relevant, and its use more efficient. It also allows a feedback to the system owner, notifying him of the irrelevance of some of the topics he has chosen.

Example 3: We use the previous lexicon (see Example 1). Topic `petrol` has been pruned, considering its proximity with `fuel` lower that the threshold δ_s.

IV. User attractivity & Querying

A. User Data Acquisition

From the source of data we analyze, we retrieve the various publications of system users. The source system may be a website, a blog, a social network, a online newspaper allowing comments, or any platform allowing users to publish content. This operation can usually be done by a wrapper specifically designed for the given source, including a specific parser and outputting data in a normalized format. One can also rely on classical API to extract information such as the Facebook API.

B. Publication attractivity

For each analyzed content, we look for extract main topics and for each one, define the publication attractivity by topics. We use the previously pruned lexicon $(t_1, \ldots, t_n)$ containing all topics relevant to search. If there are n topics in the lexicon, we can figure that each topic t is assigned a dimension in vector space, then the lexicon is the basis of a n-dimensional hypercube. Every publication p may be thought as a topic vector in this space, so $\bar{p} = (p_{t_1}, \ldots, p_{t_n}) \in \mathbb{R}^{+^n}$.

To determine the topics addressed in a publication, we use a derivative work of Das et al. [4]. This method involves analyzing the document with five different algorithms to determine with a simple majority if the text contains a feeling about the topic (positive or negative), or not relevant at all. As we consider the interest and not the opinion, we interpret both feelings as a positive vote as interest. Based on it, we build a vector for each publication.

This method, using five different algorithms and a base dataset initialized by the Web Analyst, has the advantage of providing relevant and reliable results by not raising the content that does not win the majority. In other words, quality over quantity analysis of information extracted. As an interesting side effect, it also allows us to eliminate spam messages. They did not win the majority of tests, and they are simply ignored.

Example 4: Considering the previous lexicon (see Examples 1 and 3), the publication "`Review of my new Carrera`" will be mapped to:

$$p = (0, 5, 0, 0, 3, 1).$$

This means that the topic `Porsche` is considered highly relevant (Carrera is the name of car series build by Porsche). The topic `engine` is identified as a topic with average importance inside the document, and `fuel` as a minor topic. Topics `Ferrari` and `dealership` are considered non-relevant from the document.

C. User Attractivity

Based on all collected publication attractivity, we are now able to compute the user attractivity as a vector of the same type as previously. We define this u vector, with $u \in \mathbb{R}^{+^n}$, such as $u = \sum p$ with p being publication of the user.

We use a sum rather than normalizing these results, in order to maintain the independent nature of the rate of involvement. For example, if a user is the author of numerous contributions related to a given topic, normalizing the results would reduce its importance in this topic community if it publishes many documents in another independent topic. It makes no sense in this case.

D. Graph

We now have a semantic graph of the lexicon (see Section III), and a vector attractivity u per user. We translate these vectors into semantic distance, as follows:

$$\textsc{Dist}(u, t_i) = \frac{1}{\log\, u_{t_i}}$$

Finally, users are positioned on the graph, according to their attractivity.

Example 5:

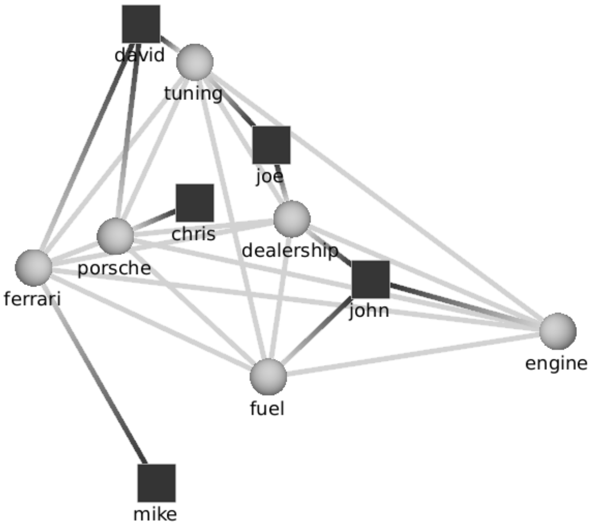


Figure 3. Example of Topic Graph after user positioning

E. Querying Communities

After users have been positioned semantically, it is possible to group them according to given parameters. By "parameters" we mean the choice of one or more subjects, with their logical operators if necessary, and a threshold.

An user u is considered as a member of the community of topic t_i, if $u_{t_i} < \delta_C$, where δ_C is a threshold set by the Web analyst.

This method allows to dynamically build communities, and adjust the threshold according to the needs of the query.

Example 6: According previous lexicon, the Web analyst can perform boolean queries, such as

$$(\text{dealership} \cap \text{engine}) - \text{Ferrari},$$

that selects user talking about dealership and engine, but not for Ferrari.

F. Viewing

The results of previous queries are nodes of the graph, with weighted relations between them, based on semantic distances. This allows to represent the community resulting from the query as a graph, which can be visualized by the web analyst and exploited by him.

G. Incremental Issue

1) New publication: The system is planned for a continuous crawling of the targeted sites, and a scalable analysis. For example, when a target receiving new messages, they must be added to the analysis. Because the formula of the semantic distance between a user and a topic is invertible, we do not need to store user attractivity vectors (see above). For each new publication, it is just needed to extract all of its attractiveness topic. For each topic t_i evaluated with an attractivity a, we update the semantic distance between the user u, author of the publication, and the topic t_i as follows:

$$\text{DIST}'(u, t_i) = \frac{1}{\log\left(10^{\frac{1}{\text{DIST}(u,t_i)}} + a\right)}$$

2) New topic: For various reasons (policy, new behaviors occurrence, etc.) the Web analyst may need to add new topics to the lexicon. Then, if the new topic is relevant (see Section III), we locate it in the topic graph as usual. After that, we have to evaluate the distance between all users and it. If the whole log of old publications is memorized, we compute the publication attractivity the new topic for each publication, and define a new semantic distance for users as usual. If we do not have archives, we approximate the new distances. As we know the semantic distances between the old topics and new one, we evaluate the distance from each user to the new topic as the value of the shortest path between them.

V. Project Milestones

We extracted several thousands of user comments to USA Today [5], an U.S. online newspaper. All these contributions are signed by their authors, who are identifiable (authenticated users). This extraction was performed by a wrapper specifically developed for USA Today, including HTML and JSON parsers. All contributions are stored as standard XML documents.

Early versions of our semantic graphs have been produced in GML format for viewing. Our graph visualizations have been produced with Tulip [6]. We plan to implement a SQL storage, to take into account the transitivity problems of a system operating in real time.

To develop the use of the system by the web analyst, we plan to define a social query language, which performs logical operations (union, intersection, complement, etc.) on the semantic graph.

VI. Related work

Since the Web birth until now, the community concept has evolued. Many works propose different approaches, depending on whether we consider a community as a set of Web pages, or as a group of people sharing a topic of interest.

Discoverning Web Communities

Since the early work on discoverning Web communities [7], hyperlink is used as a discovery basis. a major contribution in this regard is the Kleinberg HITS algorithm which defines the notions of authorities and hubs, structuring a community [8].

Imafuji et al. [9] define a page as member of a community if this page is more referenced from inside the community than outside. They use a maximum flow algorithm to isolate the nodes belonging to a community, based on the algorithm proposed by Flake et al. [10].

Dourisboure et al. [11] then identify, within a Web graph, communities as many dense bipartite sub-graphs in this graph. The bipartite graph represents for one side the interests of the community (according to the authorities HITS) and for the other side those who cite the community (the hubs). This method identify possible sharing of similar interests in different user communities, or rather the sharing of the same user group in different topic communities.

These approaches provide an advanced link analysis between pages, making topic communities, but however do not to bring users to their interests or activities: the hyperlink sharing is no longer necessarily the basis of the exchanges of the collaborative Web (content evaluation by the user, tags, etc.).

Semantic Distance

Cattuto et al. [12] propose another statistical approach for evaluating semantic distances. They validated it on data from the `del.icio.us` [13] website. This website has community structure, and the authors use the annotation data to construct a weighted network of resources. In this context, the similarity between resources is proportional to the overlap of their set of tag, representing a topic. To take into account the tag representativeness, the TF-IDF method is used. The authors propose to detect communities of users by the similarities of their tags. They use the Pearson correlation coefficient as similarity measure, and then apply methods of partitioning. As they do not reduce the number of tags handled, the tag set may be extremely large.

Recommendation Systems

The topic combination is also used in the recommendation systems. By defining the system *Socialranking*, Zanardi et al. [14] do an enrichment query based on tag similarity, based themselves on their common appearances on different resources. Another approach is proposed by Hotho et al. [15] under the name *FolkRank* and again using the graph theory. This approach use *PageRank* to model the relationships between resources, users and tags. This approach, which more exploits the sparse relations, is also explored by Bertier et al. [16] under *Gossple*. The authors use the probability of moving from one tag to another as an indicator of their similarity. Dziczkowski et al. [17] propose a recommendation system based both on the automatic analysis of uses (activity) and profiles written by users. Their method emphasizes the importance of linguistic classifier in understanding the user. This is one reason why we chose the mixed solution of Das et al. [4].

VII. Conclusion

In this paper, we have presented a complete system based on the analysis of user publications. We extract communities that depend on common interests of those users, based on their activities. The communities generated are depending of Web analyst query, validating the fact that there are no absolutes communities, but communities on application.

In order to provide an experimentation, this work will be extended so that social interactions between users are extracted, based on, for, example, forums threads. We plan to developp a complet tool that will allow the Web analyst to fully discover and exploit his communities, as explained on this paper.

References

[1] F. Abbattista, M. Degemmis, N. Fanizzi, O. Licchelli, P. Lopes, G. Semeraro, and F. Zambetta, "Learning user profiles for content-based filtering in e-commerce," in *AI*IA 2002: Proceedings of the 8th Congress of the Italian Association for Artificial Intelligence*, 2002.

[2] R. Carreira, J. M. Crato, D. Gonçalves, and J. A. Jorge, "Evaluating adaptive user profiles for news classification," in *IUI '04: Proceedings of the 9th international conference on Intelligent user interfaces*. New York, NY, USA: ACM, 2004, pp. 206–212.

[3] R. Cilibrasi and P. M. Vitanyi, "Automatic meaning discovery using google," in *Kolmogorov Complexity and Applications*, ser. Dagstuhl Seminar Proceedings, M. Hutter, W. Merkle, and P. M. Vitanyi, Eds., no. 06051. Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2006.

[4] S. R. Das and M. Y. Chen, "Yahoo! for amazon: Sentiment extraction from small talk on the web," *Management Science*, vol. 53, no. 9, pp. 1375–1388, 2007.

[5] "Usa today news," http://www.usatoday.com/news/, (visited 01/06/2010).

[6] D. Auber, "Tulip : A huge graph visualisation framework," in *Graph Drawing Softwares*, ser. Mathematics and Visualization, P. Mutzel and M. Jünger, Eds. Springer-Verlag, 2003, pp. 105–126.

[7] D. Gibson, J. Kleinberg, and P. Raghavan, "Inferring Web communities from link topology," in *HYPERTEXT'98: Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems*. New York, NY, USA: ACM, 1998, pp. 225–234.

[8] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," in *SODA'98: Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1998, pp. 668–677.

[9] N. Imafuji and M. Kitsuregawa, "Effects of maximum flow algorithm on identifying Web community," in *WIDM'02: Proceedings of the 4th international workshop on Web information and data management*. New York, NY, USA: ACM, 2002, pp. 43–48.

[10] G. W. Flake, S. Lawrence, and C. L. Giles, "Efficient identification of Web communities," in *KDD'00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2000, pp. 150–160.

[11] Y. Dourisboure, F. Geraci, and M. Pellegrini, "Extraction and classification of dense communities in the Web," in *WWW'07: Proceedings of the 16th international conference on World Wide Web*. New York, NY, USA: ACM, 2007, pp. 461–470.

[12] C. Cattuto, A. Baldassarri, V. D. P. Servedio, and V. Loreto, "Emergent community structure in social tagging systems," *Advances in Complex Systems (ACS)*, vol. 11, no. 04, pp. 597–608, 2008.

[13] "del.icio.us," http://delicious.com, (visited 01/06/2010).

[14] V. Zanardi and L. Capra, "Social ranking: uncovering relevant content using tag-based recommender systems," in *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*. New York, NY, USA: ACM, 2008, pp. 51–58.

[15] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme, "Folkrank : A ranking algorithm for folksonomies," in *LWA 2006: Lernen - Wissensentdeckung - Adaptivität, Hildesheim, October 9th-11th 2006*, 2006, pp. 111–114.

[16] M. Bertier, R. Guerraoui, V. Leroy, and A.-M. Kermarrec, "Toward personalized query expansion," in *SNS '09: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*. New York, NY, USA: ACM, 2009, pp. 7–12.

[17] G. Dziczkowski, L. Bougueroua, and K. Wegrzyn-Wolska, "Social network - an autonomous system designed for radio recommendation," *Computational Aspects of Social Networks, International Conference on*, vol. 0, pp. 57–64, 2009.

Semantic-based Technique for the Automation the 3D Reconstruction Process

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*—The reconstruction of 3D objects based on point clouds data presents a major task in many application field since it consumes time and require human interactions to yield a promising result. Robust and quick methods for complete object extraction or identification are still an ongoing research topic and suffer from the complex structure of the data, which cannot be sufficiently modeled by purely numerical strategies. Our work aims at defining a new way of automatically and intelligently processing of 3D point clouds from a 3D laser scanner. This processing is based on the combination of 3D processing technologies and Semantic Web technologies. Therefore, the intention of our approach is to take the human cognitive strategy as an example, and to simulate this process based on available knowledge for the objects of interest. First, this process introduces a semantic structure for the object description. Second, the semantics guides the algorithms to detect and recognize objects, which will yield a higher effectiveness. Hence, our research proposes an approach which uses knowledge to select and guide the 3D processing algorithms on the 3D point clouds.**

Keywords - Semantic web; knowledge modeling; ontology; 3D processing; mixed strategy; 3D scene reconstruction; object identification

I. Introduction

The laser scanning technology is a powerful tool for many applications; it has partially replaced traditional surveying methods since it can speed up field work significantly. This results in rich datasets with lots of useful and useless information. On one hand, the "manual" processing of such data set is efficient and robust since a human uses his own knowledge for detecting and identifying objects in point clouds, but this process is tedious, time-consuming and expensive. On the other hand, the "automatic" processing of 3D point clouds can be very fast and efficient, but often it relies on significant interactions with the user for controlling algorithms and verifying the quality of the results. The WiDop project [24] aims at the automatic processing of 3D point clouds using the specialist knowledge in order to guide the reconstruction process. By this way, the point clouds quantification and qualification will not be processed via an intermediary step allowing the human intervention (Figure 1). The principle of the WiDop project is a knowledge-based detection of objects in point clouds for AEC (Architecture, Engineering and Construction) engineering applications.

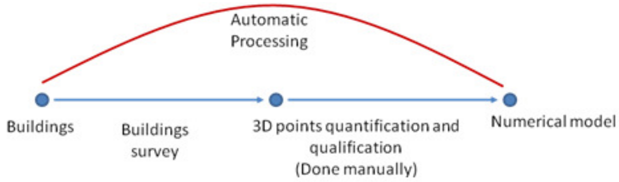


Figure 1. Automatic processing compared to the manual one.

Funded by the German government, the partners of the WiDop project are the German railway company (Deutsche Bahn), the Fraport company (Frankfurt Airport manager), and the Metronome company specializing in 3D point cloud processing. The Fraport company main concerns are building and furniture management of the airport. The furniture's position relative to the security gates and the trashes are constantly moving. In addition, updates are done on buildings such as new walls, destruction of walls, new holes in a wall, new windows, etc. This could be undertaken by the director of a new shop or by the technical employers in order to reorganize storerooms for instance. As a matter of fact, it is very difficult to keep up to date the plans of the airport. The motivation of the Deutsche Bahn Company is the management of railway furniture. The issue is closed to the Fraport Company because they have to face the management of the furniture which changed constantly. The cost of keeping these plans up to date is increasing. The solution consists to fix on a locomotive a 3D terrestrial laser scanner and to survey the surrounding landscape. After the first survey, the resulting data will be considered as a reference for comparisons with future surveys in order to detect changes. As a consequence, both companies will benefit from an automatic processing, because too much data has to be processed, and the amount of data leads to a tremendous management cost.

Ten years ago, a new format, which seems to be very suitable for our purpose was developed by the IAI (The International Alliance for Interoperability) it is named the IFC format (IFC - Industry Foundation Classes). The specification is a neutral data format to describe exchange and share information typically used within the building and facility management industry. This norm considers the building elements as independent objects where each object is characterized by a 3D representation and defined by a semantic normalized label. Consequently, the architects and the experts are not the only ones who are able to recognize the elements, but everyone will be able to do it, even the system itself. For instance, an IFC door is not just a simple collection of lines and geometric primitives recognized as a door; it is an "intelligent" object door which has a door attributes linked to a geometrical definition. The building mock-up for instance is designed by engineers and it describes all concrete and abstract elements of a building. Thus, it allows each participant in a building project to share and exchange information with the standardized description. IFC files are made of objects and connections between these objects. Object attributes describe the "business semantic" of the object. Connections between objects are represented by "relation elements" [1]. This format and its semantics are the keystone of our solution.

The following section covers background information on works and projects that aim at the reconstruction of 3D scenes from 3D point clouds. Section 3 presents a summary of the designed solution. Section 4 describes in detail the WiDop project. Section 5 focuses on the general model conception and the interaction management between the different created layers. It gives an overview of the different components of the reconstruction process and the basic theory of the "WiDop mixed strategy" presented by the combination of semantic web technology and 3D processing algorithms. Finally, conclusions and suggestions for future work are presented.

II. RELATED WORK

The reconstruction of 3D scene covers a wide area of computer vision; such reconstruction is based on the 3D processing algorithm extracted from the signal processing domain. Recent works aims to reconstruct a scene based on semantic networks describing the relationship between the scene objects. Based on these observations, this section will be articulated in two parts: the first one presents reconstruction methods based on signal processing algorithms while the second one describe methods based on semantic networks technology. Within a photogrammetric domain, there are three classes of methods for 3D scene reconstruction: Manual, semi-automatic and automatic methods.

A. *3D Processing Methods*

Within the Terrestrial Scanning Laser (TSL) processing, three different main method classes are identified. These methods are classified based on their automatic rate. This section is articulated in three parts. The first part presents reconstruction methods based on manual processing of 3D point of clouds. The second presents the semantic based method to assist in the 3D scene reconstruction process and finally, the third one shows the automatic processing methods.

1) Manual methods: are completely based on user interactions. Such methods allow the user to extract the scene elements, which are then converted into 3D models with the help of software's packages.

2) Semi-automatic methods: in these methods, the user initializes the process by some manual measurements based on which an algorithm tries to extract other elements. Such methods are based on user interactions and automatic algorithm processing. They support elements projection, affine, and Euclidean geometries [2] for the definition of constraints. When modeling buildings by constructive solid geometry, buildings can be regarded as compositions of a few components with simple roof shapes (such as flat roofs, gable roofs and hip roofs). In [3], Vosselman et al. tried to reconstruct a scene based on the detection of planar roof faces in the generated point clouds based on the 3D Hough transform. The used strategy relies on the detection of intersection lines and height jump edges between planar faces. Once done, the component composition is made manually.

3) Automatic methods: these methods are processed without the need of any kind of user intervention. Manual methods have been established with the appearance of the need to reconstruct 3D scene long time ago and are available under a high end commercial feature c.f. Leica® [18] or low cost software Dista [16]. Automatic methods use various approaches but all are based on segmentation techniques to extract features. The methods of Pollefeys et al. [4] and Zisserman et al. [5] use the projective geometry technique. Pollefeys method divides the task of 3D modeling into several steps. The system combines various algorithms from computer vision, like projective reconstruction, auto-calibration and depth map estimation. The disparity calculation between point pairs makes it possible to get a depth map. The depth map is then transformed into a volume model composed of voxels. The surface estimation between the outer surface voxels and the interior surface voxels makes it possible to combine inner and outer object parts. The method developed is effective and obtains good results. The approach of Zisserman et al. [5] proceeds in two steps. First, a coarse surface model of the building is carried out. Then the coarse model guides the search of details (windows and doors) and refines the surface model. The reconstruction uses the detection of "vanishing points", line correspondence, and the estimation of points and homologous lines. Vanishing points are necessary for the detection of planar primitives with the help of the plane-sweeping method. This method has strong constraints as it contains three perpendicular dominant directions.

B. *Knowledge-based Methods*

All the strategies outlined below are based on signal processing algorithms, in the other side, new strategies appeared recently. They are based on semantic networks to guide the reconstruction like the work of Cantzler et al. [9], it aim to improve the structural quality of a 3D model. Architectural features like orientation of wall are used. Then, the feature's relationships are automatically extracted using a semantic network of the building mock 'up. The whole strategy consists of three steps: the architectural feature extraction from triangulated 3D model. Then the automatic extraction of constraint out of the scene is carried out by matching the planes against a semantic network of the building mock 'up by backtracking research tree. In this step, the semantic network concentrates the definition of the 3D objects and the relationships between them. The constraints such as parallel or perpendicular wall are exploited. The last step consists in applying the constraint to the model. Consequentially, the original model will be fitted to the new constraint model. Ansgar et al. [6] presents a new concept for the building reconstruction. Building model is reconstructed based on it is topology using Markov model technique. Stephane et al. [7], investigates this work into a model based reconstruction of complex polyhedral building roofs. The roof in question is modeled as a structured collection of planar polygonal faces. The modeling is done into two different regimes, one focus on geometry, whereas the other is rules by semantics. Concerning the geometry regime, the 3D line segments are grouped into planes and furthers into faces using a Bayesian analysis. In the second regime, the preliminary geometric model is subject to a semantic interpretation. The knowledge gained in this step is used to infer missing parts of the roof model (by invoking the geometric regime once more) and to adjust the overall roof topology.

C. *Discussion*

The problem of automatic object reconstruction remains a difficult task to realize in spite of many years of research [8]. The major problems are the impact of the viewpoint onto the appearance of the object, resulting in changes with respect to geometry, radiometry, and existence of occlusions and the lack of texture. Strong variations in the viewpoint may destroy the adjacency relations of points, especially when the object surface shows considerable geometrical variations. This dissimilarity causes confusions within correspondence determination and is even worse, when partial occlusions result in a disappearance of object parts. In cases of weak texture, algorithms do not have sufficient information to correctly solve the correspondence problem. Consequently, the reconstruction fails to give a solution. Cantzler et al. [9] and Nüchte et al. [10] tried to solve these problems by using semantic information coupled to a scene.

Planes found in the reconstruction phase are introduced into a semantic interpretation, which has to fit to a network model [11]. A tree of "backtracking" allows the finding of the best mapping between the interpretation of the scene and the semantic network model. A coherent labeling exists, if all surfaces are labeled. Relations between the nodes of the semantic network are used to define geometrical constraints between labeled surfaces. The model used and the relations between the elements of the model define the knowledge of a typical architectural scene. The interpretation of the scene then forms a semantic network, which is an instance of the architectural model. Actually, we argue that the pervious cited works and others do not take in account the context of the geometries, and the use of 3D processing algorithms. Based on these observations, the idea behind this work is to benefit from the knowledge related to the scene structure and the different characteristics of geometries mainly to select the most suitable 3D processing algorithm from a 3D processing algorithm collection. In addition, in order to resolve the ambiguities issue of the scene caused by the sited constraints, more than one interaction between the semantic network and the 3D point clouds data is required.

In this paper, we claim that the domain of the semantic Web, and semantics technologies that it relies on, is of benefit for the definition of an automatic processing. One of the technologies is a language that helps to define ontologies; an evolved version of the semantic networks. Ontologies presents one of the most famous technology for knowledge modeling, where the basic ideas was to present information using graphs and logical structure to make computers able to understand and process it easily and automatically [12]. Our approach aims to structure knowledge, link geometrical objects to semantic information, create rules and finally guide the algorithms selection in 3D point clouds processing. The created knowledge will be structured in ontology. The produced ontology to orient the 3D object identification contain variety of data like the GIS data, images capture synchronized with the point clouds, information about the objects characteristics, the hierarchy of the sub elements, the geometrical topology, different processing algorithms etc. In the automatic process, the modeled knowledge will provide to the system relevant information aiming to orient the localization and the identification process. This purpose is reached by selecting the most suitable algorithm for the object detection and recognition. To achieve it, the ontology must contain information about objects characteristics like positions, geometrics information, images textures, etc. and also about the most suitable detection algorithms for each of existent objects.

III. SYSTEM OVERVIEW

As mentioned above, the automatic processing of 3D point clouds can be very fast and efficient, but often relies on significant interaction of the user for controlling algorithms and verifying the results. Alternatively, the manual processing is intelligent and very precise since a human person uses its own knowledge for detecting and identifying objects in point clouds, but it is very time-consuming and consequently inefficient and expensive. If human knowledge could be inserted into automatic detection and reconstruction algorithms, point cloud processing would be more efficient and reliable. However, such a solution involves a lot of questions and challenges such as: (1) How can knowledge be structured based on heterogeneous sources? (2) How to create a coarse model suitable for different applications? (3) How to allow a dynamic interaction between the knowledge model and the 3D processing part?

In general, mathematical algorithms contain different data processing steps which are combined with internal decisions based on numerical results. This makes processing inflexible and error prone, especially when the data does not behave as the model behind the algorithm expects. We want to put these implicit decisions outside, make a semantic layer out of it and combine it with the object model. This approach is more flexible and can be easily extended, because knowledge and data processing are separated.

The created knowledge will serve to guide the numerical algorithms for 3D point cloud processing, based on rules that have been created and formalized before. The knowledge will be organized in an ontology structure. Knowledge not only describes the information of the objects, but also gives a framework for the control of the strategies selected. For instance, it provides rules for the localization and identification process. These rules guide the selection of individual algorithms or sequences allowing the detection and recognition of the object to be searched for. Once the knowledge provides initial information about the structure of the scene and the objects, candidate regions can be determined. Then, the algorithms integrated in the knowledge will be guided to identify objects. In other cases, when the existence of objects in the scene is ambiguous, we will search them in the point cloud based on updated information in the knowledge model. Consequently, knowledge-based methods will enable the algorithms to be executed reasonably and adaptively on particular situations. This is where WiDOP project will try to make a step forward.

IV. WIDOP PROCESSING CAPACITY

The WiDop project aims at the development of efficient and intelligent methods for an automated processing of terrestrial laser scanner data. Figure 2 presents the general coarse architecture for the WiDop project, composed of 3 parts: the knowledge part, the 3D processing part and the interaction management and control part labeled (WiDop Processing) ensuring the interaction between the above sited parts. In contrast to existing approaches, we aim at the utilization of previous knowledge on objects. This knowledge can be contained in databases, construction plans, as-built plans or Geographic Information Systems (GIS). Therefore, this knowledge is the basis for a selective, object-oriented detection, identification and, if necessary, modeling of the objects and elements of interest in the point cloud.

A. *The knowledge processing*

Our approach aims at structuring and modeling the existing knowledge in order to represent objects from the geometrical and the semantic point of view and to integrate important feature characteristics, if necessary. In the second step, this knowledge base will guide the numerical algorithms for 3D point cloud processing, based on rules that have been created and formalized before. This approach also follows the concept of Semantic Web, while the knowledge will be organized in an ontology structure, where the basic idea is to present information in a logical structure to make computers able to understand and process it easily and automatically.

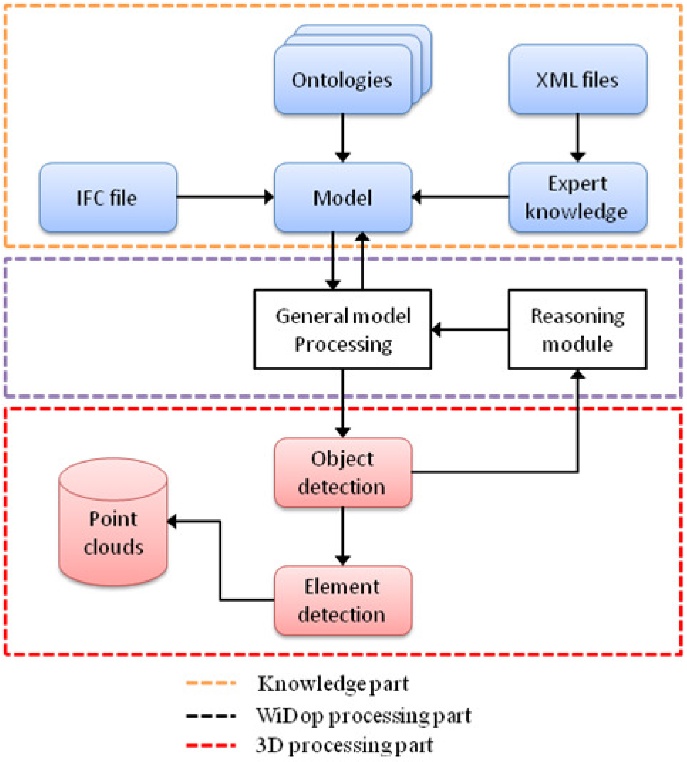


Figure 2. Overview system

Our approach is intended to use semantic knowledge based on OWL technology for knowledge modeling and processing. Knowledge has to be structured and formalized based on IFC schema, XML files, etc., using classes, instances, relations and rules. An object in the ontology can be modeled as presented; a room has elements composed of 4 walls, a ceiling and a floor. The sited elements are basic objects. They are defined by their geometry (plane, boundary, .), features (roughness, appearance, etc.), and also the qualified relations between them (adjacent wall, perpendicular, etc.). The object “room” gets its geometry from its elements and further characteristics may be added such as functions in order to estimate the existent sub elements. For instance a “classroom” will contain “tables”, “chairs”, “a blackboard”, etc. The research of the object “room” will be based on an algorithmic strategy which will look for the different objects contained in the point cloud. This means, using different detection algorithms for each element, based on the above mentioned characteristics, will allow us to classify most of the point region in the different element categories. This prior knowledge is modeled in a Coarse Model (CM). It corresponds to the spatial structure of a building and it is an instance of semantic knowledge defined in the ontology. This instance defines the rough geometry and the semantics of the building elements without any real measurement. For example, a CM may define the number of stages, the type of roof, the configuration of the walls, the number of rooms per floor, the number of windows and doors per wall. In a CM, images and point clouds may be used as entry parameters for the process of data collection trying to correct the CM.

B. *The 3D processing:*

Numerical processing includes a number of algorithms or their combination to process the spatial data. Strategies include geometric elements detection (straight line, plane, surface, etc.), projection - based region estimation, histogram matrices, etc. All of these strategies are either under the guidance of knowledge, or use the previous knowledge to estimate the object intelligently and optimally. Alongside with 3D point clouds various types of input, data sets can be used such as images, range images, point clouds with intensity or color values, point clouds with individual images oriented to them or even stereo images without point cloud. All sources are exploited for application to particular strategies. Knowledge not only describes the information of the objects, but also gives a framework for the control of the selected strategies. The success rate of detection algorithms using RANSAC [21], Iterative Closest Point [22] and Least Squares Fitting [23] should significantly increase by making use of the knowledge background. However, we are planning not only to process point data sets but also based on a surface and volume representation like mesh and voxels, respectively. These methods will be selected in a flexible way, depending on the semantic context.

C. *The WiDop processing:*

In order to manage the interaction between the knowledge part and the 3D processing part, a new layer labeled the WiDop processing is created. This layer ensures the control and the management of the information transaction and the decision taken, based on several steps, as outlined in Figure 3. The steps are:

- The algorithmic strategy selection.
- The update of the *Coarse Model.*
- The topological search of new objects.
- The semantic characterization of new objects.

In the next section, the mixed strategy based on the WiDop processing layer is presented in detail in order to show the different interactions that takes place during the WiDop reconstruction process between the knowledge base and the 3D processing algorithms.

V. THE INTERACTION MANAGMENT

We propose a mixed strategy based on WiDop processing layer insuring the interaction between the knowledge base and the 3D processing algorithms (Figure 2). It presents an intermediary between the semantic based strategy and the 3D processing one. In this section, the global view of our mixed strategy will be presented and the ultimate interaction between both of parts is described.

As seen in Figure 3, the mixed strategy is based on two principles axes which are the geometric resolution based on the 3D processing domain and the semantic one based on the semantic web technology.

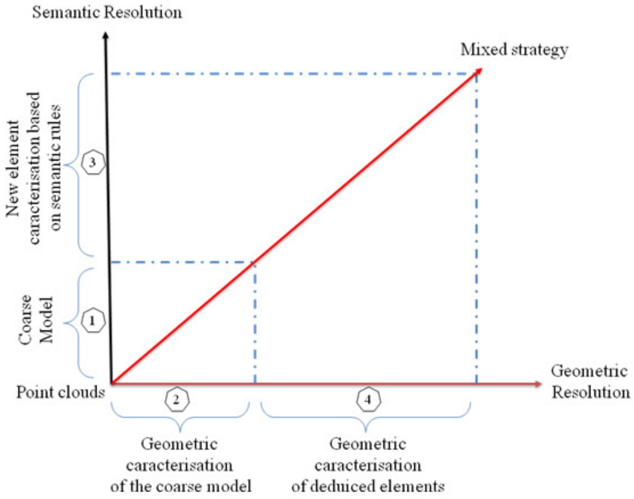

Semantic Resolution
Mixed strategy
New element caracterisation based on semantic rules
3
Coarse Model
1
Point clouds
2
4
Geometric Resolution
Geometric caracterisation of the coarse model
Geometric caracterisation of deduiced elements

Figure 3. The mixed strategy, a system overview.

Such strategy can be divided in two main steps: The first step is the geometric quantification, detection and recognition of the different existent objects in the coarse model. In this phase, the purpose of the processing is the detection of the defined objects in the coarse model. This is ensured by linking the high level semantic object definition in the coarse model and the correspondent portion of the point clouds. The second one aims at the semantic characterization of new objects in point clouds is based on the topologic relations. The inference will be based on the "Coarse model" CM and on the detected and localized objects. In this step, based on the relation´s interpretation and the interference rules management, new objects in the point cloud will be inferred and detected automatically (Figure 3).

In order to focus on our method for the combination of the semantic web technology and the 3D processing algorithms, Figure 4 illustrates an UML sequence diagram that represents the general design of the proposed solution. Hence, the purpose is to create a more flexible, easily extended approach where algorithms will be executed reasonably and adaptively on particular situations. The system architecture is divided into four actors: the data base, the 3D processing, the WiDop processing and the knowledge base.

To simplify the illustration, we will use a single data set type. In fact, we are limited in working on point clouds generated by a laser scanner. This does not mean that we will not profit from others resources like images, panoramic images, videos, etc. For this reason, the mentioned source presenting the fourth actor in the diagram is a laser scanner providing millions of point clouds. For the rest of the section, the real mechanism related to our solution will be disambiguated in details beginning by how our ontology is created, how knowledge are linked to the 3D processing algorithms arriving to how objects are detected and semantic model is updated.

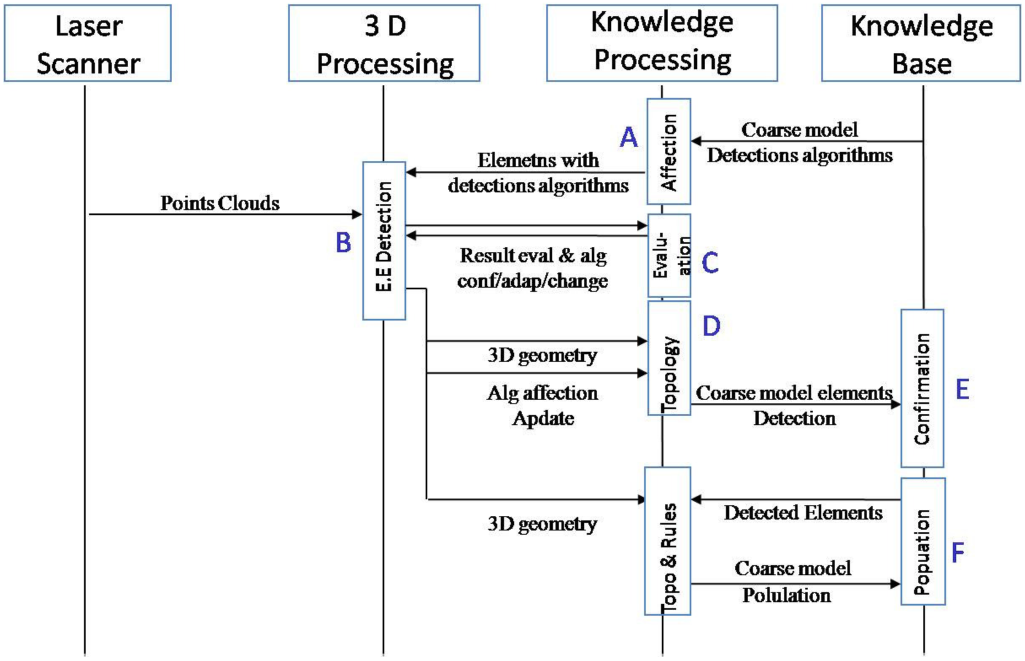


Figure 4. The sequence diagram of interactions between the laser scanner, the 3D processing,the knwoldege processing and the knowledge base.

A. *The Ontology Creation*

The WiDop project deals with the creation of an ontology corresponding to the project requirements. In this field, two different strategies for the ontology creation can be used. In the first one, the ontology is created manually depending on our vision and on the business knowledge provided by the specialists of the domain. Such ontology will look like a bottom up ontology [13], [14], very precise and designed for a specific domain. In the second one, it can be automatically generated based on different sources like ontologies from different domains such as the transport, the railway and the geometric ones [15]. The generation of the ontology can be also done based on software's packages thanks to many tools like the XML2OWL [17], [19]. It serves to map XML files provided from Metronome Db Clear Suite software used for the management of the Deutsche Bahn point cloud´s, allowing a manual tagging of the different selected elements and describing the general structure for the railway domain to an OWL file. It can also be ensured by the IFC/XML tools mapping IFC files for the building management structure to OWL one. From our point of view, the WiDop ontology must respect the applied areas specification (railway or Fraport). Based on this observation, our ontology is created automatically in order to have a general model then adapted manually to respect the real scene characteristics. The schema extracted from the XML data base provided from the DB Clear suite software, will be exploited to facilitate the automatic population of our ontology. Once our knowledge base is created and populated, it will be used as an entry for the WiDop project (Figure 5).

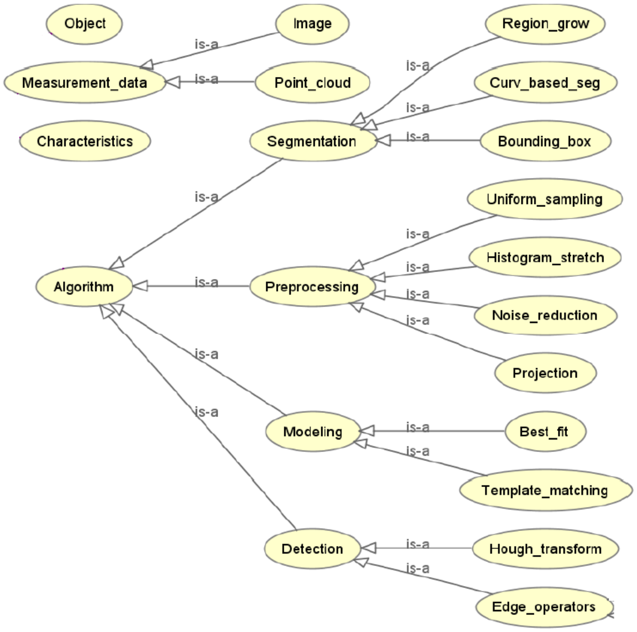


Figure 5. Portion of the developped ontology describing the "Algorithm" class

B. *Integrating Knowledge in 3D object detection*

The proposed approach couples the semantic web technology represented by the knowledge to the 3D processing one represented by the 3D processing algorithms. Let's remember that the idea behind this project is to direct, adapt and select the most suitable algorithms based on the objects characteristics. In fact, one algorithm could not detect and recognize different existent objects in the 3D point clouds, since they are distinguished by different shapes, size and capture condition. The role of knowledge is to provide not only the object's characteristics (shape, size, color, etc.) but also object's status (visibility, correlation) to algorithmic part, in order to adjust its parameters to adapt with current situation, Table 1. Based on theses observation, we draw links from algorithms to objects based on the similar characteristics, as Figure 6 shows.

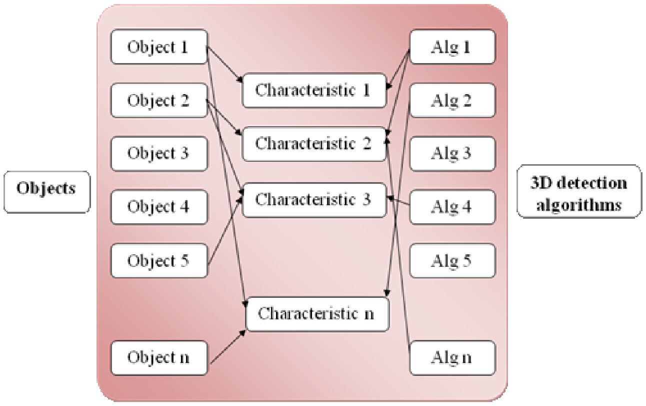


Figure 6. Linking algorithms and objects.

The knowledge part controls one or more algorithms for the detection of objects. In order to carry out this detection, we benefit from the experience of experts in 3D processing. This experience helps to find a match between the object’s characteristics and the algorithm’s characteristics. Actually, a certain algorithm can be used for the detection of a certain object in a certain context. The set of characteristics are determined by the object’s properties such as geometrical features and appearance. Then, the role of the knowledge is also to provide the algorithms that can detect and recognize these characteristics. These characteristics are considered as values and it can change the parameters of the algorithms. After the detection of an object, there is a module that gives a feedback about the status of the detected object according to the knowledge part and in order to adjust the algorithms to improve the robustness. Due to these frequent updates, the combination of knowledge and the 3D processing becomes relevant and flexible, c.f. Table1.

TABLE 1. THE CHARACTERISTICS LIST OF ALGORITHM'S AND OBJECT'S INPUT

No	*Characteristics*
1	Geometry (plane, sphere, arc)
2	Corner
3	2D boundary
4	Size
5	Orientation
6	Appearance (colour, surface material)
7	Visibility
8	Correlative position

C. *The Geometry Processing*

The third part in this model is the digital treatments. This part will focus on the object detection based on the prior knowledge and the selected algorithm. As seen in Figure 4, once the algorithms are affected, the 3D processing layer will provide the generated point clouds from the laser scanners, it will also be provided with the different pieces of information relative to each object in the ontology. The 3D layer must have as information:

- The object label
- The object location coordinate
- The object spatial coordinate
- The eventual 3D shape of the object
- The sub-elements composing the object
- The object complexity rate
- The most suitable detection algorithm to use
- …….

Depending on the object complexity, there are two possible scenarios. In the case of a low complexity rate, the objects can be detected automatically based on a template matching algorithm [20]. Else, the objects will be decomposed into elementary sub-objects as shown in the area B of Figure 4. Once detected, an evaluation process will estimate the detection quality rate (Figure 4, area C), and a topological reconstruction of the root element will be executed (Figure 4, area D). Once the coarse model elements are detected and recognized, (Figure 4, area E), the semantic research of new objects step is stimulated.

D. *The Semantic Qualification*

The described technique for the geometric qualification of the coarse model above aims at the detection of the maximum existing elements in the CM. Normally, a real scene should always contain extra object or unexpected one. To ensure a high detection quality rate, we suggest a second main module for our strategy aiming to identify new object in the coarse model. In fact, knowledge contains a reasoning capacity able to infer logical consequences from a set of asserted facts. Our model will be able to infer new objects and relations based on the coarse model topological relation and on the detected and identified elements (Figure 4, area F).

VI. CONCLUSION AND FUTURE WORK

The proposed approach for 3D object recognition in point clouds labeled "Mixed Strategy" present our initial work and vision on the project. It aims to improve the object localization and the scene reconstruction leading to a more robust and efficient processing of 3D point clouds and image data since it is based on interaction between two complementary domains, the semantic web and the 3D processing one via an intermediary layer labeled WiDop processing.

The integration of knowledge into 3D processing is a promising solution. It could make the object detection algorithms more robust, flexible and adaptive in the different circumstances through the knowledge guidance via a new mechanism under construction named "3D processing rules". Such mechanism aims to connect ontology to 3d processing algorithm via new Built-Ins. Once executed, these rules will query the ontology and the point clouds via the activation and the instantiation of the most suitable 3D processing algorithm.

ACKNOWLEDGMENT

The work presented in this paper is part of the research project WiDOP – Wissensbasierte Detektion von Objekten in Punktwolken für Anwendungen im Ingenieurbereich, funded by the German Federal Ministry for Research and Education (grant no. 1758X09). Partners in the project are Metronom Automation GmbH and DB Netz AG (for the railway domain) and Fraport AG (for the facility management domain).

REFERENCES

[1] Vanlande, R., Nicolle, C., and Cruz, C. 2008. “IFC and building lifecycle management,” Elsevier, Automation in Construction, Vol. 18, pp. 70-78.

[2] Zitova, B., and Flusser, J. 2003. “Image registration methods: a survey,” Elsevier, Image and vision computing. Vol. 21, pp. 977-1000.

[3] Vosselman, G., and Dijkman, S. 2001. “3D building model reconstruction from point clouds and ground plans,” International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, Vol. 34, pp. 37-44.

[4] Pollefeys, M. 2000. “Automated reconstruction of 3D scenes from sequences of images,” Elsevier, ISPRS Journal Of Photogrammetry And Remote Sensing, Vol. 55, pp. 251-267.

[5] Hartley, R., and Zisserman, A. 2003. “Multiple view geometry in computer vision,” Cambridge University Press New York, NY, USA.

[6] Brunn, A. 2000. “A step towards semantic-based building reconstruction using markov-random-fields,” INTERNATIONAL ARCHIVES OF PHOTOGRAMMETRY AND REMOTE SENSING, v 33, pp. 117-124.

[7] Scholze, S. Moons, T., and Van Gool, L. 2002. “A probabilistic approach to building roof reconstruction using semantic labelling,” Pattern Recognition. pp. 257-264.

[8] Grimson, W E. 1986. “From images to surfaces: A computational study of the human early visual system,” MIT press Cambridge, Massachusetts.

[9] Cantzler, H. Fisher, R., and Devy, M. 2002. “Quality enhancement of reconstructed 3D models using coplanarity and constraints,” Springer, Pattern Recognition, pp. 34-41.

[10] Nuchter, A., Surmann, H., and Hertzberg, J. 2003. “Automatic model refinement for 3D reconstruction with mobile robots,” IEEE Computer Society, pp. 394-401.

[11] Grau, O. 1997. “A scene analysis system for the generation of 3-D models,” First International Conference on Recent Advances in 3-D Digital Imaging and Modeling (3DIM '97) pp. 221.

[12] Borst, WN., Akkermans, JM., and Top, JL. 1997. “Engineering ontologies,” International Journal of Human-Computer Studies, Vol. 46, pp. 365-406.

[13] Van der Vet, PE., and Mars, NJI. 1998. “Bottom-up construction of ontologies,” IEEE Transactions on Knowledge and data Engineering, Vol. 10, pp. 513-526.

[14] Hare, JS., Sinclair, P. A. S., Lewis, P. H., Martinez, K., Enser, P. G. B., and Sandom, C. J. 2006. “Bridging the Semantic Gap in Multimedia Information Retrieval: Top-down and Bottom-up approaches,” Mastering the Gap: From Information Extraction to Semantic Representation / 3rd European Semantic Web Conference, 12 June 2006, Budva, Montenegro.

[15] Stumme, G., and Maedche, A . 2001. “Ontology merging for federated ontologies on the semantic web,” Proceedings of the International Workshop for Foundations of Models for Information Integration (FMII-2001), pp. 413-418.

[16] Dista: http://www.i3mainz.fh-mainz.de/Article68.html. The last access date: 04-08-2010.

[17] Bohring, H., and Auer S. 2005. “Mapping XML to OWL ontologies,” Leipziger Informatik-Tage, Vol. 72 of LNI, GI, pp.147-156.

[18] Leica®: http://www.leica-geosystems.fr/fr/index.htm. The last access date: 04-08-2010.

[19] Cruz, C., and Nicolle, C. 2008. “Ontology Enrichment and Automatic Population From XML Data,” ODBIS. pp.17-20.

[20] Kenue, S.K. “LANELOK: Detection of lane boundaries and vehicle tracking using image-processing techniques- part II: Template matching algorithms,” SPIE Conference on Mobile Robots. pp. 234-245.

[21] Lacey, AJ., Pinitkarn, N., and Thacker, N.A. “Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation,” Proceedings of the British Machine Vision Conference (BMVC) 2000.

[22] Besl, P.J., and McKay, N.D. 1992. “A Method for Registration of 3-D Shapes,” IEEE Trans. Pattern Analysis and Machine Intelligence, V 14, pp. 239-256.

[23] Arun, KS., Huang, T.S., and Blostein, S.D. “Least-squares fitting of two 3-D point sets,” IEEE TRANS. PATTERN ANAL, MACH. INTELLIG 1987. pp. 698-700.

[24] Marbs, A., Boochs, F., Ben Hmida, H., and Truong, H. 2010. “Wissensbasierte Objekterkennung in 3D-Punktwolken und Bildern,” DGPF-Tagungsband, 3-Ländertagung D-A-CH Conference Wien, pp. 220-227.

Semantic Process Modeling and Planning

Michael Igler
Chair for Applied Computer Science IV
University of Bayreuth
Bayreuth, Germany
michael.igler@uni-bayreuth.de

Stefan Jablonski
Chair for Applied Computer Science IV
University of Bayreuth
Bayreuth, Germany
stefan.jablonski@uni-bayreuth.de

Christoph Günther
Chair for Applied Computer Science IV
University of Bayreuth
Bayreuth, Germany
christoph.guenther@uni-bayreuth.de

***Abstract*—In this paper, we investigate how complex process models can be modeled such that both the modeling remains doable for domain experts and the resulting process models remain readable. We chose an approach that can be characterized as mixed approach consisting of declarative and imperative modeling aspects with semantically enriched process modeling constructs. Our aim is to benefit from both modeling approaches, declarative and imperative, whereby through their combination we want to avoid their drawbacks. However, the implementation of our modeling approach is completely declarative. In this paper we present our novel approach for process modeling and together with its implementation. Some experiences about how domain users are applying this approach are also given.**

***Keywords*-Semantically enriched process modeling constructs; flexible process execution; process planning.**

I. Introduction

Process management has been accepted as adequate method to describe complex business applications and to support their enactment. Deliberately we focus on complex applications since there the benefits of a process-based approach are of particular importance. Process models illustrate nicely how complex applications are structured and describe what has to be done by which person using which tools. However, we believe that process management approaches still do not cope well with complexity. In order to substantiate this proposition we want to analyze the causes of complexity. We focus the discussion of complexity on two situations. A process-based application is complex if it consists of a huge number of different process steps (step complexity). It is not so easy to reduce this kind of complexity. Such an application can be structured by creating sub-processes through decomposition. Then process models are at least easier to understand. However, it is hard to eliminate process steps such that the application gets ”smaller”. Step complexity is a kind of an inherited feature. There is a chance that domain experts recognize that some process steps are not necessary; then this complexity can be reduced partially. A second sort of complexity arises when a huge number of execution paths exists (path complexity). In this case, the number of process steps might even be moderate. However, through the flexibility of many different execution paths complexity escalates. For example, consider three process steps `A`, `B`, and `C`

- which all have to be executed exactly once, and
- whose executions must not overlap.

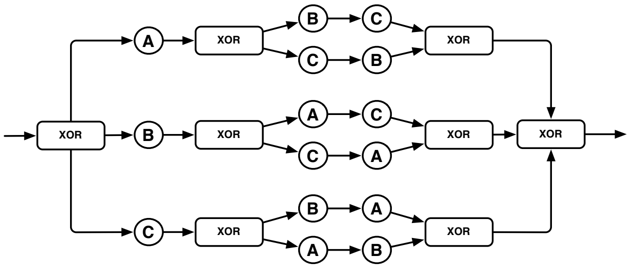


Fig. 1. Example process model

In Figure 1, a solution to this scenario is depicted. We regard this process model as complex: although only three different process steps are involved, the process model consists of 15 process steps (repetitions of the three basic steps `A`, `B`, and `C`), 29 arcs, and 8 flow constructs (XOR) for splitting and joining control flow. In this context it is not so relevant how to count steps and arcs; the message is that there are a lot of modeling elements although the application is rather small. The most severe drawback of this process model is that its pragmatics (what it means from an application point of view) is totally camouflaged, i.e., users do not comprehend the meaning and purpose of the process. We state that path complexity is partially avoidable when powerful process modeling constructs are applied.

II. Motivation

What is the reason that still path complexity is not dealt with adequately? We see one of the major reasons in the adoption of execution rules from imperative programming languages like sequential execution, alternative execution (if-then-else; XOR between execution paths) or independent execution (parallel execution paths). It is not that we blame imperative programming languages it is just that we state that this programming style is not adequate for process modeling. The fact that programs are going to become complex is not that bad since programs are just read by programmers, i.e., software experts that are able to cope with that complexity. In contrast to that process model complexity is problematic. Process models must also (besides professional process modelers) be readable for end users like medical doctors or nurses, who are usually not so familiar with formal process modeling techniques. Thus, when process models are becoming too complex, these people cannot interpret them anymore. That also means that they cannot assess their quality anymore and therefore cannot improve them. As a consequence we really want to promote applying process modeling techniques, which reduces complexity such that complex applications can be described by comprehendible process models and can therefore be understood much more easily. We propose to apply declarative process modeling techniques that specifically reduce path complexity. In contrast to imperative modeling (here the path(s) going through a process is defined explicitly) declarative modeling concentrates on describing what has to be done and the exact step-by-step execution order is not directly prescribed.

In order to provide another motivating example we introduce a second scenario which would result in a most complex process model if it is only described with conventional process modeling elements (such that are borrowed from imperative programming). The (simplified) scenario originates from a clinical study. Four process steps are involved: `Take blood sample` (from now on called `A`), `Measure intraocular pressure (B)`, `Measure blood pressure (C)`, and `Write report (D)`. It is not a problem to model this scenario with conventional means under the assumption that the four steps should be executed sequentially and each step must be executed exactly once. However, if it is allowed to execute `A`, `B`, and `C` in any order (but not overlapping) then the model adopts the complexity of the process model in Figure 1 and is presented in Figure 6. Complexity increases again when process steps `A`, `B`, and `C` could be executed multiple times (if their execution results are not satisfactory). Finally we would like to restrict the number of executions for each process step `A`, `B`, and `C` individually. Process step `A` should be executed once or twice, process step `B` doesnt have to be executed at all but can be performed once; `C` must be executed once and can be repeated arbitrarily. We abstain from listing the possible execution sequences. It becomes obvious that a huge number of such sequences could be produced. It is also a challenging exercise to define process models for the two last extensions of our application scenario.

Our goal is to introduce process modeling constructs that facilitate to model complex and flexible processes in a compact and comprehensible way. Our first idea was to switch from imperative modeling (as it is quite common in process management) to declarative modeling (see Section IV). However, studying [1], which discusses the pros and cons of imperative and declarative process modeling, we decided that a complete switch to declarative process modeling is not optimal, since that would mean to abandon the good aspects of imperative modeling. Thats why we have chosen an approach that tries to combine the pros of both approaches: declarative and imperative. Thus, we can reduce the obstacles of both modeling approaches as well. So, our modeling approach is a combination of declarative and imperative concepts. However, the underlying implementation of our process execution engine is a purely declarative one.

Besides this main requirement we want to post two further requirements, which are vital for our approach. We state that the semantics of these new process modeling constructs must be unique, i.e., they must be precisely defined. This requirement refers to both process modeling and execution. We explicitly name this requirement since we will allow a great degree of freedom for process execution. This looks like allowing arbitrary execution orders. However, this is not the case as we will discuss in Section IV. The third major requirement focuses process execution. Since we deal with complex scenarios that should be described by compact process models, end users should be guided through the execution of such processes, i.e., there should be a possibility to highlight recommended execution paths among the many eligible execution paths. Hence we give the process executors (e. g. nurse, medical doctor) some guidance through the process flow but still sustain the flexibility in choosing the next process step according to the actual personal perception of the process executors involved.

III. Related Work

The current approaches to process modeling can be categorized as either imperative approaches or declarative approaches. In this section, we compare our work with some representative implementations of both imperative and declarative approaches.

A. Imperative approaches

Wohed et al. [2] made an evaluation of the suitability of the imperative modeling language BPMN [3]. It evaluates the modeling languages against the workflow patterns from [4] concluding that there exist inherent difficulties in applying a language that does not have commonly agreed-upon formal semantics nor an execution environment. Although there is a mapping from BPMN to the execution environment BPEL [5], closer inspections show that this mapping is only partial, leaving aside models with unstructured topologies.

The research described in [6] does a comparison of business modeling and execution languages coming from the open source area. It concludes that open source systems like jBPM [7] and OpenWFE (now called ruote 2.1) [8] are geared more towards modelers who are familiar in programming languages than towards business analysts.

YAWL (Yet Another Workflow Language) [9] is a workflow language that make use of so called high-level Petri nets to refer to Petri nets extended with color, time and hierarchy. The definition of YAWL presented in [9] only supports the control-flow (behavioral) perspective. Therefore newYAWL [10] has been developed to provide support for the control-flow, data and resource perspectives. Nevertheless the functional and organizational perspective are still neglected.

B. Declarative approaches

DECLARE [11] is a constraint-based system, developed at the University of Eindhoven, that is focused on modeling constraints between processes. It supports the behavioral and the functional perspectives of Perspective Oriented Process Modeling method (POPM) [12]. DECLARE uses the ConDec [13] modeling language. Modeled constraints in ConDec are translated to a Linear Temporal Logic (LTL) formula. There is an automaton generated for every specific constraint in order to verify it. Furthermore, an automaton is also generated over all constraints. The support for the organizational perspective in DECLARE is, however, limited as hierarchical structures cannot be modeled. A planning component that can be consulted for advice during execution phase is also absent.

EM-BrA^{2}CE (Enterprise Modeling using Business Rules, Agents, Activities, Concepts and Events) is a framework for unifying vocabulary and execution models for declarative process modeling [14]. The vocabulary is described in terms of the Semantics for Business Vocabulary and Rules (SBVR) standard and the execution model is presented as a Colored Petri Net (CP-Net). EM-BrA^{2}CE also follows the same concept we use in this paper to specify a state space transition relation based on rules. However, functional and operational perspectives are not supported in this framework. Furthermore, the process modeler has no possibility to graphically “model” business process. Instead, every process must be described in the form of the mentioned Business Vocabulary. This slows down re-reading of process models by different users or the process modeler itself after some period of time.

IV. Introducing New Elements for Compact Process Modeling

This section presents three new modeling elements, which form the basis of our approach. Since we focus on the reduction of path complexity we introduce three new declarative modeling elements: special arrows (with two different semantics), boxes (to group processes), and quantification (to define the number of executions of a process). Besides these new modeling constructs we rely on the typical modeling elements of the perspective oriented process modeling method [15]. However, in this paper we mainly focus on the functional perspective and the behavioral perspective, whereas we neglect the data, operational and organizational perspectives.

A. Two Different Types of Arrows

The first modeling construct that will be associated with a new semantics is the arrow. The semantic of the well known arrow symbol in process modeling is that if an arrow connects process A with process B then process B has to be performed after process A. Accordingly, if process B is connected with an arrow to process C then C may start after process B has finished (Figure 2). We also say: B requires the execution of A before it can run; C requires the execution of B (and consequently of A, too) before it can run. We want to keep this very common construct and put it in our modeling toolbox. We present this modeling construct as a solid line.

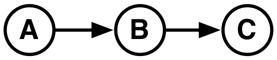


Fig. 2. Sequential process flow

Beside this arrow construct depicted by a solid line we want to add an arrow depicted by a dashed line; this dashed arrow holds a different meaning. Two processes that are connected through a dashed arrow can be executed in any order. For instance, if process A and process B are connected by a dashed arrow, then A can be performed before B or vice versa (B can be performed before A). Along with this construct we see the optimal combination of the imperative and the declarative approaches. Applying imperative concepts would require much more arrows and deciders to express the flexibility; this would blow up the process model drastically and lead to unreadable models; exclusively applying declarative concepts would avoid any arrows and the ”natural” understanding of a process flow would be lost. Furthermore, having defined a dashed arrow from process A to process B expresses a preference (recommendation) that process A should be performed before process B. This feature can be utilized when processes are put on a work list for execution. If more than two processes are connected through a dashed line then a permutation of all process executions is feasible, e.g., ABC, BCA, CBA. Processes must not be performed in parallel but sequentially. Modeling the scenario from Figure 1 using the dashed arrow, results in the simple process model of Figure 3. It is obvious that the process model of Figure 3 is much more readable than the process model of Figure 1.

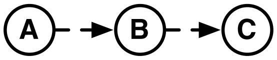


Fig. 3. Model of flexible scenario

It is certainly possible to combine the solid and dashed arrows: In Figure 4 process A and B are connected through a dashed arrow; process B and process C are connected through a solid arrow. This means that there is flexible ordering between processes A and B while process B must always be executed before process C. This semantics results in the following three execution orders: ABC, BAC and BCA.

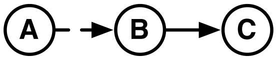


Fig. 4. Combination of solid and dashed arrows

Now consider the medical example from Section I. To connect the three process steps A, B, and C with dashed arrows offers to execute them in an arbitrary order. However, so far it is not possible to say that process step D must be executed after the three process steps A, B, and C have terminated. In order to express this semantics a new modeling element has to be introduced, which is called box.

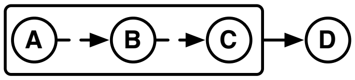


Fig. 5. Box with final acceptance process D

B. The Box Modeling Element

The box modeling element ensures that all the processes inside a box are regarded as a unit. Thus a box can substitute a process. That means that instead of executing a single process A or B the box must be performed, that means the processes within the box must be executed. For instance, in Figure 5 the box must be executed completely before process D can be started. Executing the box means to execute processes A, B, and C in an arbitrary order. This execution results in the following sequences: ABCD, BCAD, CABD, CBAD, ACBD and BACD. D is always the last step that requires the completion of all previous steps respectively the box, in which the steps are contained.

The process of Figure 5 clearly models the medical scenario from Section I, when processes A, B, and C must be performed in any order before process D.

C. Quantification

Often it is necessary to specify that a process can be executed several times. For that purpose we add quantificational aspects to process steps which are novel in the field of process modeling. Every process gets a minimum and maximum counter that indicates how often a process may be executed. If it shall be executed exactly a certain number of times then minimum and maximum are equal. To express that a process step is not essential for the whole process but can be done in the sense of “possible but not necessary”, then a minimum quantification of zero should be selected.

Reconsider the medical example of Section I. We now want to declare how often these processes may be executed:

- Process A: minimum = 1, maximum = 2
- Process B: minimum = 1, maximum = 1 (exactly once)
- Process C: minimum = 0, maximum = * (optional, any repetition)
- Process D: minimum = 1, maximum = 1 (exactly once)

Modeling this scenario by just using conventional, i.e., imperative modeling elements results in a process model as depicted in Figure 6. It is almost impossible to derive the above defined semantics from that diagram. That situation changes when our innovative combination of declarative and imperative modeling constructs is applied. Figure 7 presents the same process model as Figure 6. It is obvious that the complexity of the process model is completely vanished through the use of the new powerful modeling constructs. Users regard processes A, B, and C as a unit that has to be executed before process D can be performed. Also the flexibility of executing the processes within the box can be recognized easily. The number attached to the processes shows how often processes have to be executed. This example nicely shows that the declarative process modeling constructs facilitate compact modeling of complex process-based applications. This modeling style is advantageous for business process modeling in such a way that it is now possible to describe very complex business process models in a more elegant and easier to comprehend way.

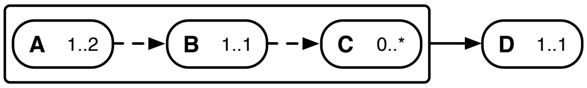


Fig. 7. Simplified medical example

Nevertheless, it is not that easy for novel users to apply our new modeling constructs. Therefore, we were directly supporting them during modeling. We experienced that after a couple of modeling sessions they were able to apply the new constructs independently and, finally, that it was even easier for them to express their complex application scenarios. Without any doubt by introducing our new modeling constructs we are leaving the realm of standards (e.g., BPMN). However, this is not an unusual approach. In many publications (see the BPM conference series [16]) new modeling constructs are introduced, which are not covered by a standard like BPMN. Many researchers and especially practitioners accept that in special cases standards have to be violated in order to provide more adequate modeling capabilities. The tradeoff between standard conformance and enhanced expressiveness has to be resolved individually for each project.

V. Architecture and Implementation

In this section, we want to give an overview on our implementation for process management, i.e., process modeling and execution. In Figure 8 the architecture of our process management infrastructure is depicted.

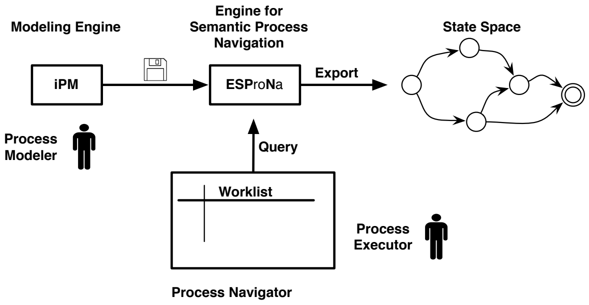


Fig. 8. Architecture of the process modeling and execution framework

Processes are modeled with the tool “iPM Process Modeler” [17]. This tool supports modeling constructs of POPM and all the modeling elements introduced in Section IV. A process model is saved in a special data format and is loaded into the planning component of the process execution system. We call this prototype ”Engine for Semantic Process Navigation” (ESProNa). A process executor (e.g., nurse, medical doctor)

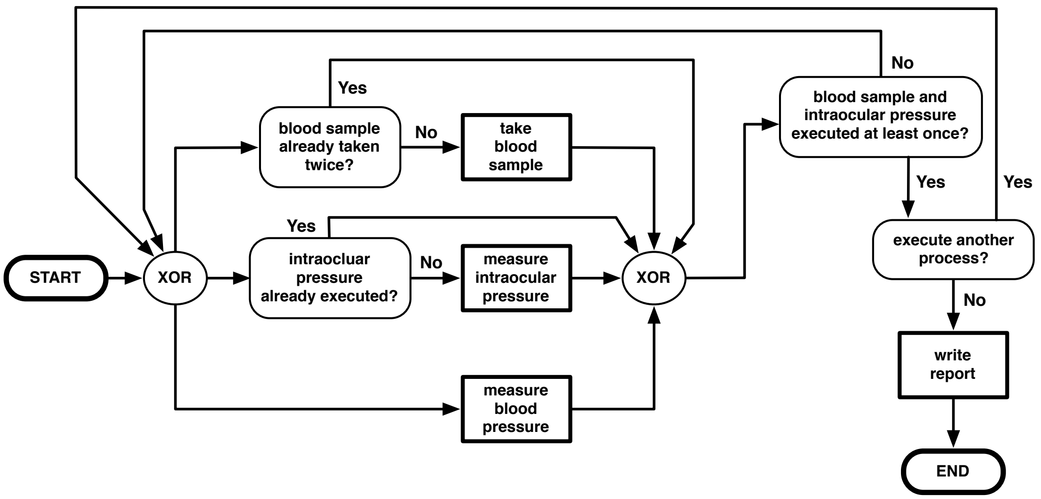
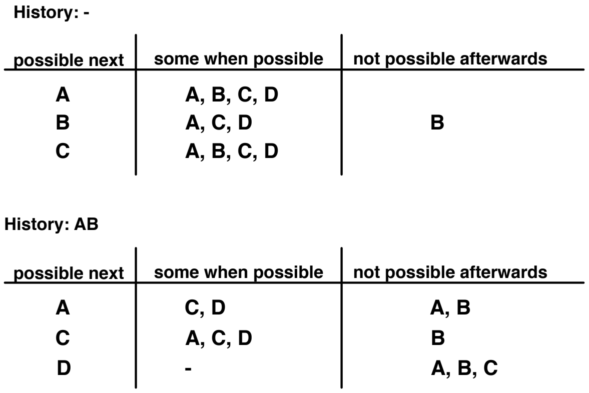

START
XOR
blood sample already taken twice?
Yes
No
take blood sample
intraocluar pressure already executed?
Yes
No
measure intraocular pressure
measure blood pressure
XOR
blood sample and intraocular pressure executed at least once?
No
Yes
Yes
execute another process?
No
write report
END

Fig. 6. Conventional modeling of medical example

can then use the Process Navigator PN [18] to navigate through a process. PN is a process execution system that works on process models deploying the Perspective Oriented Process Modeling method. A work list depicts all processes that are executable. The process executor can then select one of the executable processes to perform it. So far the PN cannot interpret process modeling elements as presented in Section IV. Our prototype ESProNa extends PN in order to cope also with these declarative modeling constructs.

A conventional process management system [19] cannot support a look ahead to the process executor such that he can see what processes are not executable anymore and what processes are still executable. This functionality is additionally provided for PN by ESProNa. With this "look ahead" some kind of guidance is provided to the process executor since he can better anticipate the impact of the execution of a special process. It is of enormous importance when flexible execution is provided as described in Section I and IV. Through the many different alternative execution paths that become available a process executor might get overburdened with the selection of processes for execution. Therefore, this guidance functions is very important and sustains a better overview on process execution. Thus we can say: introducing this highly flexible execution semantics, which drastically reduces path complexity (Section I) comes with costs: loss of overview since often very many processes are executable (what is an indication of flexible execution). However, in our approach this loss is totally compensated through the provided guidance functionality, which will support the process executors to navigate through the process. How to implement guidance? We have chosen a pretty handy approach. Instead of solely offering possible next processes (for execution) we additionally offer two more columns on a work list. These columns depict the following two sorts of processes, which support a look ahead:

- Processes that can still be executed eventually after a now executable process is performed.
- Processes that never can be executed again after a now executable process is performed.


History: -

possible next	some when possible	not possible afterwards
A	A, B, C, D	
B	A, C, D	B
C	A, B, C, D	

History: AB

possible next	some when possible	not possible afterwards
A	C, D	A, B
C	A, C, D	B
D	-	A, B, C

Fig. 9. Work list for the process executor

Figure 9 depicts the implementation of this new kind of worklist. In conventional process execution systems only the left column of the two work lists (”possible next”) in Figure 9 would be supported: this column depicts the processes that are executable next. The two columns ”some when possible” and ”not possible afterwards” depicts processes, which are still executable respectively not anymore executable after a certain process is selected for execution.

Figure 9 shows two different situations whereas the example from Figure 7 is referred to. The upper work list depicts the state when nothing is done yet (history is empty ”-”). The three processes of the box are executable (`A, B, C`); `D` is not executable since first the (elements in the) box must be performed. Selecting processes `A` or `C` means that all processes `A`, `B`, `C` and (later) `D` can still be performed again. Selecting process `B` means that `B` must not be performed again since it is only allowed to be performed exactly once (quantification). The lower work list shows a situation where processes A and B were already performed: History AB. Process B is not executable any more since it was already executed and the domain constraint (B must be executed exactly once) is prohibiting this. Process D became executable since the box could be terminated and all processes of the box are performed as often as required minimally. If process A is selected it must not be performed again since it can be executed twice at most. When process C is selected, then all processes except B are executable. In the case that process D is selected no other preceding process is executable anymore. It shall be mentioned, that this behavior is exchangeable. We can adopt them to any special business process execution semantics. For example, in the former medical example processes A, B and C are not executable again since D has started and the box containing processes A, B and C must not be executed any more.

VI. Conclusion and Outlook

Two observations become noticeable when conventional process execution (PN) is extended with the ESProNa framework: Process modeling can cope with much more complex process models without enhancing complexity, i.e., especially path complexity is well coped with (Section IV). Through the powerful implementation, process executors can effectively be guided through process execution by supporting guidance in form of a “look ahead”.

Together, compact process modeling capabilities and powerful process execution guidance provides an add-on to conventional process management that is heavily requested in literature [20], [21], [22].

ESProNa is part of the ForFlow Process Navigator. This system is developed in the joint research project ForFlow [23] among 4 Bavarian Universities and about 30 industrial partners. The Process Navigator is meanwhile in prototype use in 5 partner companies, which intend to use it in productive mode.

The most important next step in our research is to integrate the operational, behavioral and data perspective of process management. We currently investigate how these aspects can be integrated into the ESProNa Framework. First steps into this research are showing that these perspectives can seamlessly be added to the concepts defined so far. However, it is obvious that these perspectives have a major impact on execution flexibility. In a future paper, we will also analyze how workflow patterns [4] can be expressed by ESProNa to show the completeness of our modeling approach.

References

[1] D. Fahland, J. Mendling, H. Reijers, B. Weber, M. Weidlich, and S. Zugal, “Declarative vs. Imperative Process Modeling Languages: The Issue of Maintainability,” in *1st International Workshop on Empirical Research in Business Process Management (ER-BPM’09)*, B. Mutschler, R. Wieringa, and J. Recker, Eds., Ulm, Germany, Sep. 2009, pp. 65–76, (LNBIP to appear).

[2] P. Wohed, W. M. P. van der Aalst, M. Dumas, A. H. M. ter Hofstede, and N. Russell, “On the Suitability of BPMN for Business Process Modelling,” in *Business Process Management*, 2006, pp. 161–176.

[3] S. White. (2004, May) Business Process Modeling Notation (BPMN) — Version 1.0. [Online]. Available: http://www.bpmn.org/Documents/BPMN_V1-0_May_3_2004.pdf

[4] W. M. P. Van Der Aalst, A. H. M. Ter Hofstede, B. Kiepuszewski, and A. P. Barros, “Workflow patterns,” *Distrib. Parallel Databases*, vol. 14, no. 1, pp. 5–51, 2003.

[5] OASIS. (2010, Mar) Web services business process execution language version 2.0. [Online]. Available: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html

[6] P. Wohed, N. Russell, A. H. M. ter Hofstede, B. Andersson, and W. M. P. van der Aalst, “Patterns-based evaluation of open source BPM systems: The cases of jBPM, OpenWFE, and Enhydra Shark,” *Inf. Softw. Technol.*, vol. 51, no. 8, pp. 1187–1216, 2009.

[7] jBPM 3.1 Workflow and BPM made practical, Chapater 9.6 Superstates. IBM. [Online]. Available: http://docs.jboss.com/jbpm/v3.1/userguide/en/html/processmodelling.html

[8] J. Mettraux. (2010, March) ruote 2.1. [Online]. Available: http://ruote.rubyforge.org/

[9] W. M. P. van der Aalst and Ter, “Yawl: yet another workflow language,” *Information Systems*, vol. 30, no. 4, pp. 245–275, June 2005. [Online]. Available: http://dx.doi.org/10.1016/j.is.2004.02.002

[10] Z. Zhou, S. Bhiri, and M. Hauswirth, “Control and data dependencies in business processes based on semantic business activities,” in *iiWAS’08: Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*. New York, NY, USA: ACM, 2008, pp. 257–263.

[11] W. M. P. van der Aalst, M. Pesic, and H. Schonenberg, “Declarative workflows: Balancing between flexibility and support.” *Computer Science — R&D*, vol. 23, no. 2, pp. 99–113, 2009.

[12] S. Jablonski, “Functional and behavioral aspects of process modeling in workflow management systems,” in *CON’94: Proceedings of the Ninth Austrian-informatics conference on Workflow management: challenges, paradigms and products*. Munich, Germany, Germany: R. Oldenbourg Verlag GmbH, 1994, pp. 113–133.

[13] M. Pesic, H. Schonenberg, and W. M. P. van der Aalst, “DECLARE: Full support for loosely-structured processes,” in *EDOC’07: Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference*. Washington, DC, USA: IEEE Computer Society, 2007, p. 287.

[14] S. Goedertier, R. Haesen, and J. Vanthienen, “EM-BrA^{2}CE v0.1: A vocabulary and execution model for declarative business process modeling,” K.U.Leuven, FETEW Research Report KBI-0728, 2007.

[15] S. Jablonski and C. Bußler, “Workflow-management: Modeling concepts, architecture and implementation,” International Thomson Computer Press, 1996.

[16] U. Dayal, J. Eder, J. Koehler, and H. A. Reijers, Eds., *Business Process Management, 7th International Conference, BPM 2009, Ulm, Germany, September 8-10, 2009. Proceedings*, ser. Lecture Notes in Computer Science, vol. 5701. Springer, 2009.

[17] S. Dornstauder, *Handbook of the iPM Integrated Process Manager*, ProDatO Integration Technology GmbH, 2005.

[18] M. Faerber, S. Jablonski, and S. Meerkamm, “The ProcessNavigator — Flexible process execution for product development projects,” *International Conference on Engineering Design, ICED’09*, 2009.

[19] F. Leymann and D. Roller, *Production Workflow: Concepts and Techniques*. Prentice Hall PTR, September 1999.

[20] P. Heinl, S. Horn, S. Jablonski, J. Neeb, K. Stein, and M. Teschke, “A comprehensive approach to flexibility in workflow management systems,” in *WACC*. ACM, 1999, pp. 79–88.

[21] R.-M. Stefanie, R. Manfred, and D. Peter, “Correctness criteria for dynamic changes in workflow systems: A survey,” *Data & Knowledge Engineering*, vol. 50, no. 1, pp. 9–34, 2004.

[22] W. Aalst and S. Jablonski, “Editorial: Flexible Workflow Technology Driving the Networked Economy,” *International Journal of Computer Systems, Science, and Engineering*, vol. 15, no. 5, pp. 265–266, 2000.

[23] Prof. H Meerkamm and Dr.-Ing. K. Paetzhold, *Bayerischer Forschungsverbund für Prozess- und Workflowunterstützung zur Planung und Steuerung der Abläufe in der Produktentwicklung*, ISBN 978-3-9808539-7-2.

Ontology-based Modeling and Inference for Occupational Risk Prevention

Alexandra Galatescu, Adriana Alexandru
National Institute for R&D in Informatics,
8-10 Averescu Avenue, Bucharest, 011455, Romania
e-mail: {agal, adriana}@ ici.ro

Corneliu Zaharia
Stefan Nicolau Institute of Virology
285 Mihai Bravu Avenue, 030304, Bucharest, Romania
e-mail: corneliu.zaharia@virology.ro

Stefan Kovacs
National Research Institute for Occupational Safety
35A Ghencea Avenue, 061695, Bucharest, Romania
e-mail: stefan_agk@yahoo.com

***Abstract*— The paper describes and motivates the use of ontologies and of an ontology-based model in a training system (under development) for the occupational risks prevention. The personalized training (for a specified context, e.g., a given activity, workplace, operator type, work machine, etc.) will be the result of the automatic discovery of the prevention documents and actions that fit the training request. The paper also sketches the basic components of the training system for risk prevention, adapted to the proposed semantic view.**

Keywords- ontologies; ontology-based modeling and inference; occupational risk prevention; e-training

I. INTRODUCTION

The paper gives an approach based on ontologies for a web system (under development) aiming at the online, fast and personalized training for occupational risks prevention. Risk prevention is a combination of disciplines necessary to reduce the risk of injuries and fatalities in any work environment. A proactive approach is the early recognition and prevention of the risk factors

Occupational risks are a category of risks that appear in work environments with a high probability of harming people or machines. Occupational risk prevention and management comply with the principles and methodology of the *risk management* (RM) process, a key process within both private and public organizations [1].

The training for occupational risk prevention should advise the operator on the health, safety, security and environmental issues related to his work. He can ask for training before or during the execution of an activity or before the use of a certain machine.

The system presented in this paper is general and adaptable to any domain with major occupational risks (industry, biology, construction, transportation, environment, agriculture, health etc.). It unifies existing methodological rules and standards for risk prevention and provides tools for the personalized training and consulting, by the dynamic and multi-criteria selection of the prevention documents and actions. It will eliminate the need for a periodical training of the employees and will diminish the costs from the poor information on the risks.

Terminology and guidelines for risk prevention. The system relies on the standard terminology proposed with ISO CD31000 [2], combined with the terminology common to several upper-level ontologies and process models.

There are several risk-related standards published by ISO and other standards bodies, as well as many proposals and principles that refer to risk management. In 2005, ISO has initiated a working group to develop a guidance standard on RM, ISO CD31000. In conjunction with this standard, the group has updated the ISO/IEC Guide 73-Risk Management – Vocabulary [3] that gives the basic vocabulary and the definitions of RM generic terms. It encourages a mutual and consistent understanding and a coherent approach to the description of the RM activities.

In Europe, the risk prevention is subject of two directives Seveso I and Seveso II [4] that establish the domain terminology, the obligations and normative documents to be elaborated regarding the large scale industrial hazards.

State of the art in software for risk prevention. There are products for the risk control in industrial environments, and domain-specific standards and software tools for RM in health, environment, insurance, finances, construction, transportation, etc. Risk prevention is automated for the security of computers, Web, networks. Security components are integrated lately with the operating systems. Ontologies are also used mainly for the security management (of assets, networks, information systems, databases, etc.). Some examples are in [5]-[9]. There is no system based on knowledge and semantics for occupational risk prevention and for training and dynamic discovery of prevention information, documents and actions.

However, [10] proposes the risk evaluation and analysis along the life cycle of the construction projects, based on ontologies and a conceptual model. They rely on a simpler reference ontology and model and have a different inference goal. Also, [11] gives an example of Web Ontology Language (OWL) [12] ontology for occupational health. And, [13] confirms the idea that a model of occupational risks is important because it describes relevant data in the context of event occurring and this data can be transformed into knowledge navigated using an intelligent search engine (similarly to the goal of the system presented in this paper).

A semantics-based approach for risk prevention. In order to benefit from semantics, the system relies on:

- A *reference ontology* that gives the basic types of taxonomies and structures for the classification and description of the risk factors, of the relationships among them, of the consequences and actions for their prevention, etc. This ontology represents the background of a *reference model* for occupational risk prevention (represented in Fig. 2 and detailed in Section 3).
- *Domain-specific ontologies and knowledge bases,* built by the specialization of the generic concepts and relationships in the reference ontology and model.
- A *query language* and *editor* for risk prevention based on the domain ontologies and reference model. The framework for the query composition based on ontologies is given in Fig. 3.
- Automatic and semantics-based inference for search and discovery of prevention documents and actions based on the risk context and requestor's preferences.

The semantics will support the interoperability of the organizations with respect to risk prevention, by common vocabularies and model. The ontology-based inference will increase the precision of the search algorithm. Also, the vocabularies and model can be dynamically extended and used in further inferences or they can be reused in other applications.

The constructs in the reference and domain ontologies comply with a subset of the constructs proposed for OWL in [12]. The constructs in the reference model comply with the basic constructs in the entity-relationship model, adapted to the use of ontologies instead of entities.

Structure of the paper. Section 2 sketches the basic components of the training system. Section 3 describes the semantic and modeling layers for the representation of the occupational risks and of the context for their occurrence. It also enumerates the basic types of inferences that will be implemented in the system and exemplifies the composition of a training query based on ontologies.

II. Components of the Ontology-based Training System for Occupational Risk Prevention

The intended system will have components distributed on two platforms (see Fig. 1):

- *Platform for the risk design,* i.e., for the risk identification, description and analysis (Fig. 1, left);
- *Platform for the risk evaluation and decision-making* on the prevention documents and actions (Fig. 1, right).

The two platforms share the repository composed of ontologies, rules, queries and documents.

The *components of the platform for the risk design* are:

- *Ontology Editor* to build (specialization or composition) ontologies or list structures in the model given in Fig. 2;
- *Rule Editor* to define or customize rules for risk prevention in the designer's organization;
- *Query Editor* to predefine or customize queries for risk prevention training.

The c*omponents of the platform for the risk evaluation and decision-making* are:

- *Query Composition and Submission Engine* to dynamically compose the training queries.
- *Model Navigator* to navigate the ontologies in the reference model in order to compose the query, as exemplified in Fig. 3.
- *Inference Engine,* called automatically after the submission of the query, in order to perform the automatic discovery of the training documents registered in the system and of the appropriate prevention actions. Besides the conditions and constraints in the query, the discovery will also rely on rules previously defined by the risk designer.
- *Query Result Generator*, called automatically by the Inference Engine, after the document discovery, in order to arrange the results.

Discovered documents can be stored either in the system repository or in the repositories/ Web servers of the organizations registered in the system. The documents can be in different formats: Web pages, Word, .pdf, .xls, etc.

The system is developed using Microsoft Visual Studio 2008 and obout Suite for ASP.Net [14]. It integrates the expression evaluator given in [15] and the interface for the rule and query editors is inspired from [16].

III. An Ontology-based Model for Risk Prevention

The system integrates three layers representing the occupational risks and the context for their occurrence and prevention: semantic, modeling and execution layers.

The ***semantic layer*** is composed of the *reference ontology* and the *domain ontologies* that give the basic vocabularies for domains with potential risks. The domain ontologies are populated by domain experts (risk designers) using the ontology editor. They are represented by:

- domain-specific *taxonomies*, i.e., hierarchies composed of concepts connected, in this system, by relationships like: *specialization or synonymy* or *composition* (part-of) or *list*-like relationships;
- *attributes* of and *constraints* upon the concepts and relationships in each ontology.

The concept attributes in any ontology can refer to external ontologies. For example, the "domain" attribute of an "activity" in Activity ontology can be selected from Domain ontology.

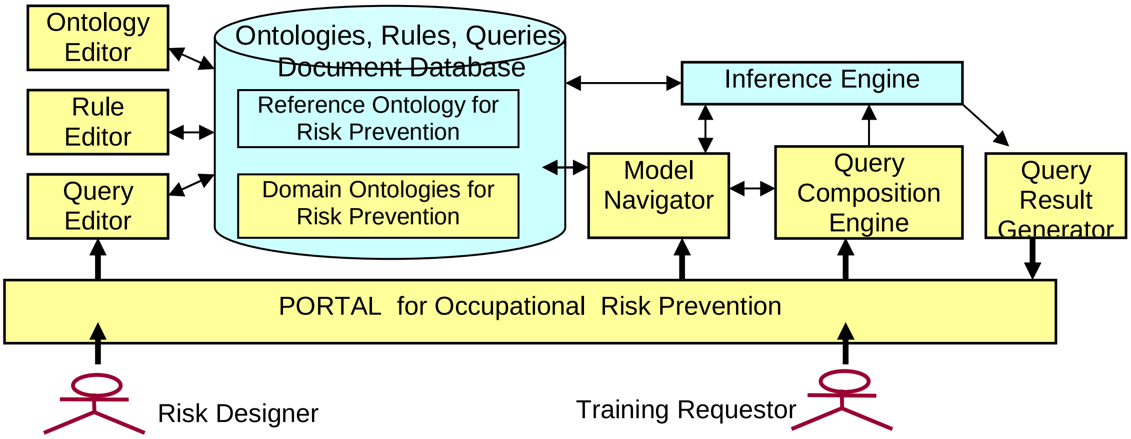

Ontology Editor
Rule Editor
Query Editor
Ontologies, Rules, Queries Document Database
Reference Ontology for Risk Prevention
Domain Ontologies for Risk Prevention
Inference Engine
Model Navigator
Query Composition Engine
Query Result Generator
PORTAL for Occupational Risk Prevention
Risk Designer
Training Requestor

Figure 1. Components of the training system for occupational risk prevention

In this system, the ontology editor treats separately the specialization, composition and list-like ontologies, because each type of ontology has its specific features. For example, the *attribute inheritance* is implicitly implemented only in the specialization ontologies. For the composition ontologies, it can be requested by the user for attributes of the ontology or of certain concepts. For list-like ontologies it is useless.

The ***modeling layer*** is needed in addition to the semantic layer in order to represent the application-specific *relationships* and *constraints* between concepts in different ontologies. In this system, the inter-ontology relationships are defined according to the reference model represented in Fig. 2. An *ontology-based model* is seen as a union of relationships between concepts in different ontologies, along with their attributes and constraints.

The ***execution (technological) layer*** represents the ontologies and models, the documents, rules, constraints and queries in a format interpretable by the software.

A. Semantic Layer for Occupational Risk Prevention

The concept types connected as in Fig. 2 and described below root ontologies based on specialization, composition or list relationships.These ontologies have been proposed to help for the identification and classification of the risk factors, of the consequences and preventive measures, of the dangerous activities and of the processes they compose, etc.

Risk**:** combination of an event probability and its consequences [3]. The event can take place in a certain workplace, during a certain activity/ task or resulting from a material source action (e.g., water, a substance, gas, etc.).

Executant **(**or Starter or Operator**)**: the (human or material) agent that, during an activity, can cause unexpected events and also can be injured by them or can get professional diseases.

Process: a sequence of activities/ operations/ tasks in a certain domain and workplace. The activities in a process can be executed by different executants, at different moments and in different places.

Activity (operation/ task): atomic operation executed independently or during a process inside the organization.

Event: occurrence or existence of a particular set of circumstances. An unpredictable event is called "incident" [3]. It can be the consequence of the executants' action using a certain instrument and acting on a certain object.

Workplace**:** location in the organization where unexpected events can occur and affect/ destroy it.

Consequence**:** outcome of an event or change in circumstances affecting the achievement of objectives [3]. An event may lead to a range of consequences. A consequence can have positive or negative effects. For the occupational risks, only the negative effects are considered.

Work_Instrument: tool/ machine/ substance/ etc., used by the operator during an activity/ task. It can determine an event or be damaged by it.

Work_Object: object existing at a workplace. It can determine an event or an event may impact on it. It can be material (e.g., a substance) or human (e.g., an infected patient in a hospital).

Document: a document containing prevention/ protection/ control instructions, regulations, rules or measures for risk prevention.

Prevention_Action: management action preventing the unexpected events or diseases. An example is the training of the operators in workplaces with potential risks.

B. Modeling Layer for Occupational Risk Prevention

Figure 2 shows how the semantic and modeling layers for risk prevention are integrated from the conceptual point of view. The modeling layer represents the relationships between the ontologies defined on the semantic layer. These relationships have been selected depending on the needed reasoning on them and on the context for the risk identification, analysis, evaluation and prevention, identified at this moment. The model can be dynamically enhanced with new ontologies, relationships, attributes and constraints that will be used in future rules, queries and inferences.

Risk modeling for their prevention and control is today mainly a mathematical modeling complemented with formal methods to assess or measure the risks and to help the decision-making for their prevention. Also, this modeling is usually a domain-specific one for health/ financial/ insurance/ economic/ business/ etc. risks.

The *benefits from an ontology-based model* in general and, in particular, for risk prevention are:

- The types of concepts and the relationships between them in the model, as well as the reasoning on them, are explicit (external to the application code) and independent of the application tools;
- The ontologies can be shared by different diagrams or models (e.g., for risk monitoring and control, in addition to risk prevention).
- The separation of the ontology-based model from the representation of the ontology content (the domain-specific hierarchies) makes it flexible, adaptable and extensible. The tools for ontology editing and navigation may differ from the tools for the model editing and navigation. Also, the reasoning on the model can be separately implemented from the reasoning on ontologies.

The basic *relationships in the reference model* are enumerated below.

Activity->Process relationship is a "*part-of*" relationship between the component activities and the process they belong to. In a process, the activities might be executed by operators in different departments and even in different organizations. The risks should be tracked for each activity, but also for each process in/ cross organizations.

Process->Process relationship is a "*part-of*" relationship between a process and its sub-processes with potential risks that should be tracked.

Activity->Workplace and ***Process->Workplace*** relationship "*executed_IN"* are necessary to track the risks per activity, process and workplace at the same time.

Executant->Activity relationship "*is_agent_of*" and also ***Activity->Work_Instrument*** and ***Executant->Work_Instrument*** relationship "*acts_WITH*" are necessary for reasoning on an *operator-activity-machine* sub-model.

Executant->Event relationship "*causes"* helps for the identification of the events that an operator might determine by his work. Inverse relationship "*acts_ON"* between ***Event->Executant*** helps for the identification of the executants that can be injured after certain events.

Work_Instrument->Event relationship "*causes*" is necessary to identify the events determined by the inappropriate use of a certain instrument. The inverse relationship "*acts_ON"* between ***Event->Work_Instrument*** is necessary to identify the instruments that can be damaged after certain events.

Besides the executants and the work instruments, the unexpected events or diseases might be caused by other objects existing at the workplace. These events can be found by the relationship "*causes*" between ***Work_Object->Event***. Also, the objects damaged by certain events can be found by the relationship "*acts_ON"* between ***Event->Work_Object***.

Event->Workplace are correlated by the relationship "*acts_ON"* in order to associate the events to the workplaces they can damage.

Risk->Event relationship "*has_effect"* associates the identified risks to the events they may produce.

Event->Consequence relationship "*has_effect"* associates the events with their consequences.

Document->Risk relationship "*describes*" associates to the identified risks the documents and the actions necessary for their prevention.

The reference model in Fig. 2 also associates the elements with potential risks that should be tracked (types of concepts like Activity, Executant, Workplace, Work_Instrument, Work_Object) with their specific risks (in Risk ontology), by the relationship "*has_risk"*.

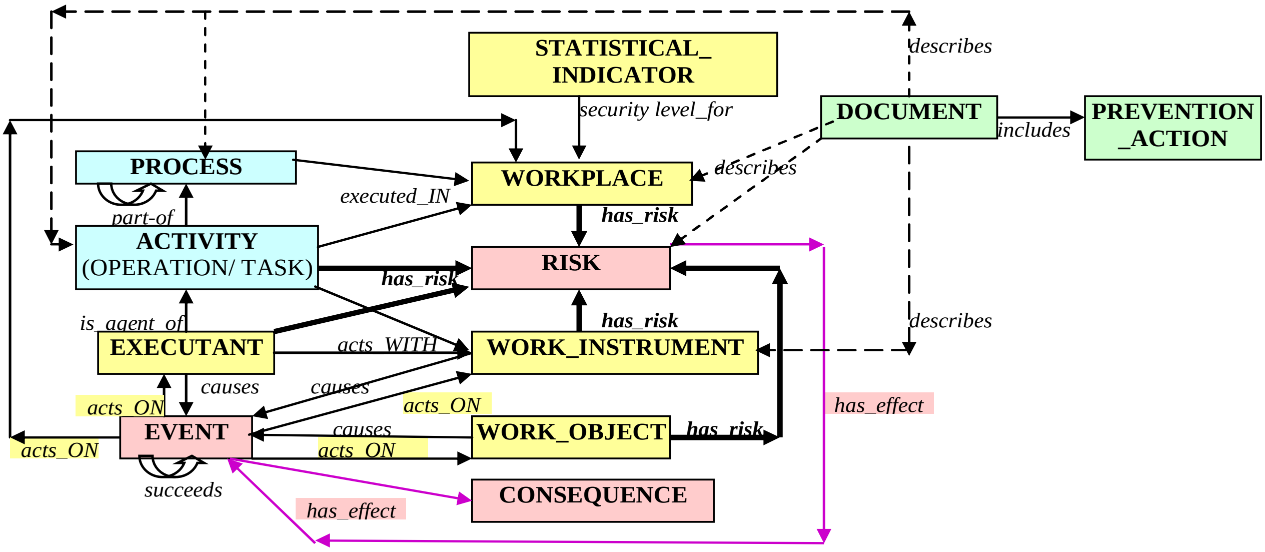


Figure 2. Basic ontologies and the relationships between them in the reference model for occupational risk prevention

The generic concepts above, their attributes, relationships and constraints in the reference model are specialized and instantiated by the risk designers, resulting in *domain models* (e.g., for biological or industrial risks). For any domain concept, the designer instantiates the concept attributes defined in the reference or domain ontology (either implicit, for the concept unique identification, or inherited or concept specific attributes). By their instantiation, the generic relationships and constraints in the reference model become domain-specific relationships between concrete concepts in ontologies.

C. Ontology-based Inference and Query Composition for Risk Prevention

The basic *knowledge and inference* for risk prevention will include (see details in the case study given in [17]):

- Rules for query formulation and for the verification of its syntactic and semantic correctness;
- Rules for the semantic completion of the search query: for each concept in a specialization ontology, its subtypes and synonyms are added in the query, as search alternatives;
- Rules for navigation on domain ontologies and models;
- Rules for risk evaluation and for search and discovery of documents and actions for risk prevention;
- Inheritance rules for both concept attributes and attribute values. In this system, for the specialization ontologies, the values of certain attributes can be inherited by concepts from their parents, at the designer's request. But, in the concept description, there are identification attributes with concept-specific values that cannot be inherited (e.g., ID, concept author, creation date).
- Inheritance rules for concept relationships. For instance, the relationship between a risk and a certain activity or workplace can be inherited by the subtypes of the respective risk.
- Rules for ordering the query results, depending on the conditions and constraints on the involved concepts.

The *query semantic completion* for a concept C is performed by the navigation in the ontology the concept C belongs to and by the extraction of its subtypes and synonyms. They are correlated with the initial concept C by the logical operator OR. Hence, the search algorithm does not use the concept C and its subtypes/ synonyms simultaneously, but successively, even if the search with the initial concept C is successful. The benefit is that more appropriate results are obtained than using only the initial concepts.

Regarding the inference for the verification of the query semantic correctness, the system will achieve:

- Verification of the *semantic compatibility between each concept type C and its concept-like instance* (value) specified in the query (when the value is a concept, not numeric). This verification is fully automated only when the value concept belongs to the same ontology as the concept type C. Otherwise, the system involves the requestor to confirm their semantic compatibility. For instance, the concept 'Laboratory_Procedure' is compatible with its instance 'p1' only whether p1 is a laboratory procedure as well, not a concept with another meaning.
- Verification of the *semantic relationships between the concepts in the query*. Suppose that the query includes two concepts Ci and Cj that belong to the ontologies Oi and Oj. Also, suppose that, previously, the designer has defined a generic relationship R between the ontologies Oi and Oj. The system checks if the designer has also instantiated the relationship R for the concepts Ci and Cj. If he did not, the occurrence of both concepts in the query might be a semantic contradiction.

  For instance, suppose that the query includes the activity *A* and the work instrument *I*. Also, suppose that between the ontologies Activity and Work_Instrument there is a generic relationship *executed_WITH*. If there is no concrete instance of this relationship between *A* and *I* (*A executed_WITH* I), it is possible that the instrument *I* is incorrectly associated with the activity A in the same query. The requestor will be notified before the request execution in order to review his request. Otherwise, he can receive results about the instrument *I* that are not correlated with the results regarding the activity *A*.

Figure 3 gives an example of query composition based on ontologies. The query has three parts:

- *Search query:* a Boolean expression with concepts as operands. For example, the user asks for the prevention rules and the prevention measures to be selected from the ontology Document and for the physical risks at the workplace to be selected from the Risk ontology);
- *Search condition:* an expression with known concepts as operands. They indicate the work context where the risks and events can occur (e.g., the activity “Laboratory_Procedure” selected from Activity ontology and the work instrument “Substance_with_microorganisms” selected from Work_Instrument ontology).
- *Concept restrictions:* expressions with concept attributes as operands. For example, the search should find the physical risks with high gravity and that can occur frequently.

After the query submission, it is analyzed and semantically completed, the conditions and constraints are syntactically and semantically analyzed and, then, the search algorithm is executed.

IV. Conclusions

The paper has described the conceptual and semantic framework of a system for training on occupational risks. It relies on a dedicated reference ontology and model, on domain specific ontologies and on reasoning on them, basically, for the search and discovery of registered prevention documents and actions.

Although the importance of the ontologies and of a model for risk prevention has already been revealed in the literature, there is no general software for on-line training, the goal of the system presented in this paper.

The system architecture was adapted to a semantics-based view on the risk prevention. Its *interface dedicated to the domain experts* moves the work for ontology editing and risk design, from IT experts to the domain experts. The system and its portal will contribute to a *knowledge repository for risk prevention* inside and cross organizations. It will be accessible from Web and will gradually replace the periodical training in organizations.

The *risk prevention model* described in this paper *can be dynamically extended* with new ontologies, relationships, constraints and rules, when necessary. They will be automatically considered in future inferences on the model.

The main *benefits* in this system from ontologies and from the ontology-based model for risk prevention are:

- *organization interoperability,* by common vocabularies and models on risk prevention represented in the reference ontology and model and in the domain ontologies and rules. They can be reused in other applications.
- *semantics-based inference,* by ontology and model-based verifications and executions of the rules and queries. They increase the search precision, completeness and correctness;
- *personalized queries* for training, dedicated to domain experts.

The system is partly implemented, as follows:

- The platform for the risk design is already implemented. Besides the *rule editor* and *query editor*, the designer can use general tools for:
  - *ontology editing* (for specialization, composition and list-like ontologies), with automatic inheritance of the attributes only for the specialization ontologies; and, for
  - *reference and domain model editing and instantiation.* These tools help the designers to add to the reference model: new ontologies, new inter-ontology relationships, new attributes for ontologies and relationships. And, to add to the domain model and ontologies: new concepts, new concept instances, new relationship instances. They also provide the graphical view of the models.
- The platform for risk evaluation and decision-making is partly implemented: the query composition engine and model navigator are finished; but, the inference engine and query result generator are under development.

References

[1] MethodWare, "ISO 31000: Risk Management Standard," 2009, http://www.methodware.com/iso-31000-risk-management-standard-published [visited, May 14, 2010]

[2] ISO, "ISO 31000: 2009 Risk management —principles and implementation of risk management," 2009, http://www.iso.org/iso/catalogue_detail.htm?csnumber=43170 [visited, May 10, 2010]

[3] ISO/ TMB/ WG, "ISO/ IEC Guide 73:2002 “Risk management — Vocabulary Guidelines for use in Standards,” 2009, http://www.iso.org/iso/catalogue_detail?csnumber=34998 [visited, May 10, 2010]

[4] MAHB (Major Accident Hazards Bureau), "Safety Management Systems - Seveso II," Official Publications of European Communities, Luxembourg, 1998, http://mahbsrv.jrc.it/ GuidanceDocs-SafetyManagementSystems.html [visited, May 11, 2010]

[5] S.-W. Lee, R. Gandhi, D. Muthurajan, D. Yavagal, and G.-J. Ahn, “Building problem domain ontology from security requirements in regulatory documents,” Proc. Intl. WS on Software Engineering for Secure Systems. ACM Press, 2006

[6] B. Tsoumas and D. Gritzalisi, “Towards an ontology-based security management,” Proc. Intl. Conf. on Advanced Information Networking and Applications (AINA), Vienna, Austria, 2006

[7] M. Klemen, E. Weippl, A. Ekelhart, and S. Fenz, “Security ontology: Simulating threats to corporate assets,” Proc. 2nd Intl. Conf. on Information Systems Security (ICISS), Springer, 2006

[8] A. Simmonds, P. Sandilands, and L. van Ekert, “An ontology for network security attacks,” Proc. Asian Applied Computing Conference (AACC), LNCS, vol. 3285, Springer, 2004

[9] P. Mitra, C. Pan, P. Liu, and V. Atluri, “Privacy-preserving semantic interoperation and access control of heterogeneous databases,” Proc. Symposium on Information, computer and communications security, ACM Press, 2006

[10] N. Forcada, M. Casals and A. Fuertes, "The Basis of a Decision Making Tool for Risks’ Evaluation Based on Ontologies," Proc. Intl. Conf. on Information and Knowledge Management - Helping the Practitioner in Planning and Building (CIB), Stuttgart, 2007, http://www.baufachinformation.de/aufsatz.jsp?ul=2008031001259 [visited, April 12, 2010]

[11] J. Kola, B.Wheeldin and A. Rector, "Lessons in building OWL Ontology driven applications: OCHWIZ – an Occupational Health Application," National e-Science Centre, 2007, http://www.allhands.org.uk/2007/programme/download490f.html?id=849&p=paper [visited, April 12, 2010]

[12] W3C, "OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax Oct. 2009," http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/ [visited, May12, 2010]

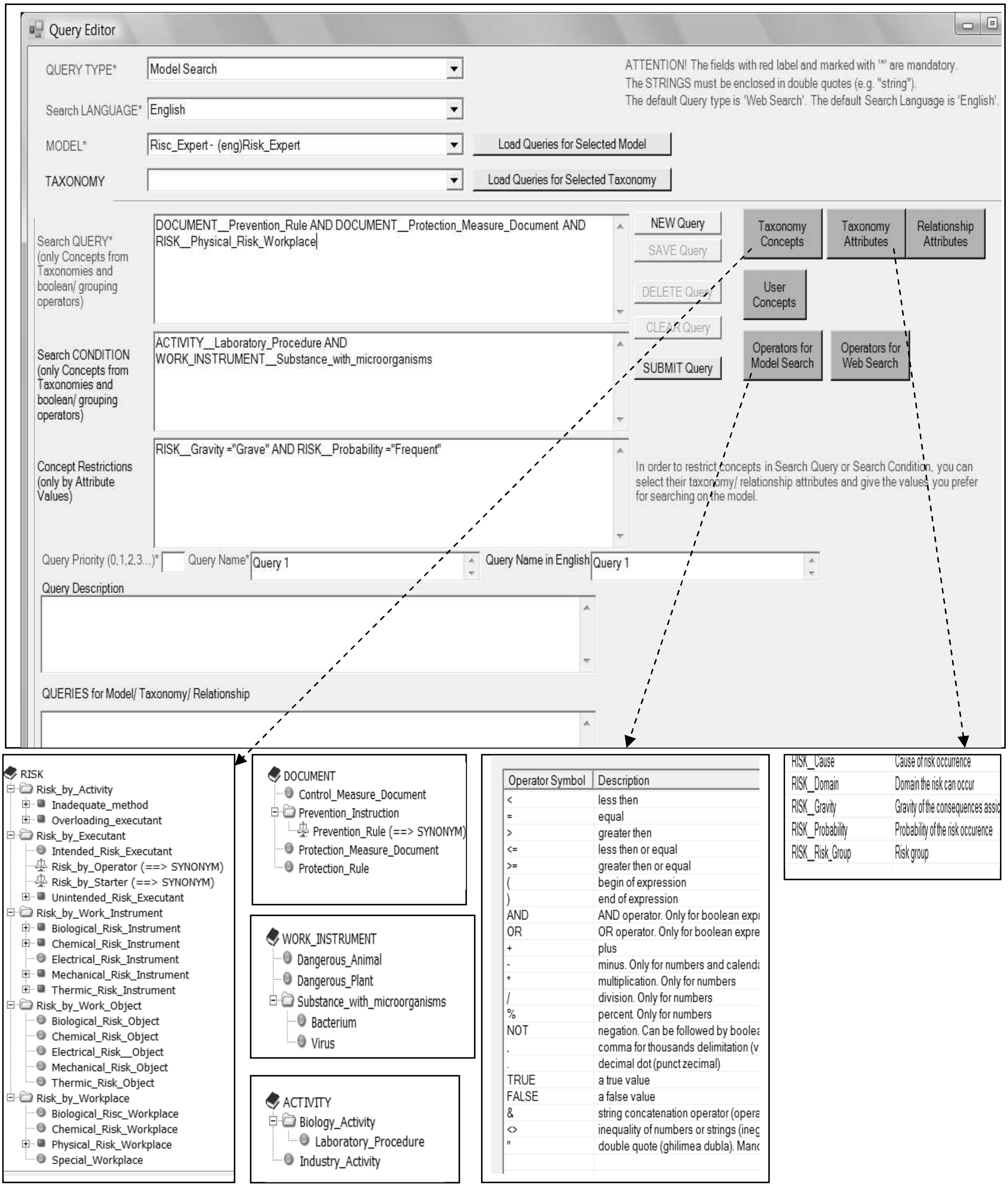
[13] P. Swuste, "Qualitative Methods for Occupational Risk Prevention Strategies in Safety or Control Banding safety," Safety Science Monitor, Issue 3, vol. 11, 2007

[14] obout,"obout Suite for ASP.Net", www.obout.com, 2010 [visited, May 20, 2010]

[15] P. Ganaye, "An expression evaluator written in VB.NET," http://www.codeproject.com/KB/vb/expression_evaluator.aspx [visited, April 2, 2010]

[16] V. Abilov, "WYSIWYG rule editor: create and test rules for any .NET type," http://bloggingabout.net/blogs/vagif/archive/ 2009/04/13/wysiwyg-rule-editor-create-and-test-rules-for-any-net-type.aspx [visited, Jan. 15, 2010]

[17] A. Alexandru, F. Filip, A. Galatescu and E. Jitaru. "Using Ontologies in eHealth and Biomedicine", in book A. Shukla and R. Tiwari (Eds) "Intelligent Medical Technologies and Biomedical Engineering: Tools and Applications", IGP Global, May, 2010



QUERY TYPE* Model Search

Search LANGUAGE* English

MODEL* Risc_Expert - (eng)Risk_Expert

TAXONOMY

ATTENTION! The fields with red label and marked with '*' are mandatory.
The STRINGS must be enclosed in double quotes (e.g. "string").
The default Query type is 'Web Search'. The default Search Language is 'English'.

Search QUERY* (only Concepts from Taxonomies and boolean/ grouping operators): DOCUMENT__Prevention_Rule AND DOCUMENT__Protection_Measure_Document AND RISK__Physical_Risk_Workplace

Search CONDITION (only Concepts from Taxonomies and boolean/ grouping operators): ACTIVITY__Laboratory_Procedure AND WORK_INSTRUMENT__Substance_with_microorganisms

Concept Restrictions (only by Attribute Values): RISK__Gravity ="Grave" AND RISK__Probability ="Frequent"

In order to restrict concepts in Search Query or Search Condition, you can select their taxonomy/ relationship attributes and give the values you prefer for searching on the model.

Query Priority (0,1,2,3...)* Query Name* Query 1 Query Name in English Query 1

Query Description

QUERIES for Model/ Taxonomy/ Relationship

- RISK
  - Risk_by_Activity
    - Inadequate_method
    - Overloading_executant
  - Risk_by_Executant
    - Intended_Risk_Executant
    - Risk_by_Operator (==> SYNONYM)
    - Risk_by_Starter (==> SYNONYM)
    - Unintended_Risk_Executant
  - Risk_by_Work_Instrument
    - Biological_Risk_Instrument
    - Chemical_Risk_Instrument
    - Electrical_Risk_Instrument
    - Mechanical_Risk_Instrument
    - Thermic_Risk_Instrument
  - Risk_by_Work_Object
    - Biological_Risk_Object
    - Chemical_Risk_Object
    - Electrical_Risk__Object
    - Mechanical_Risk_Object
    - Thermic_Risk_Object
  - Risk_by_Workplace
    - Biological_Risc_Workplace
    - Chemical_Risk_Workplace
    - Physical_Risk_Workplace
    - Special_Workplace

- DOCUMENT
  - Control_Measure_Document
  - Prevention_Instruction
    - Prevention_Rule (==> SYNONYM)
  - Protection_Measure_Document
  - Protection_Rule

- WORK_INSTRUMENT
  - Dangerous_Animal
  - Dangerous_Plant
  - Substance_with_microorganisms
    - Bacterium
    - Virus

- ACTIVITY
  - Biology_Activity
    - Laboratory_Procedure
  - Industry_Activity

Operator Symbol	Description
<	less then
=	equal
>	greater then
<=	less then or equal
>=	greater then or equal
(	begin of expression
)	end of expression
AND	AND operator. Only for boolean expr
OR	OR operator. Only for boolean expre
+	plus
-	minus. Only for numbers and calend
*	multiplication. Only for numbers
/	division. Only for numbers
%	percent. Only for numbers
NOT	negation. Can be followed by boole
,	comma for thousands delimitation (v
.	decimal dot (punct zecimal)
TRUE	a true value
FALSE	a false value
&	string concatenation operator (opera
<>	inequality of numbers or strings (ineg
"	double quote (ghilimea dubla). Manc

RISK__Cause	Cause of risk occurrence
RISK__Domain	Domain the risk can occur
RISK__Gravity	Gravity of the consequences assic
RISK__Probability	Probability of the risk occurence
RISK__Risk_Group	Risk group

Figure 3. An example for the composition, based on ontologies, of a query for training in risk prevention

GCO: A Generic Collaboration Ontology

Germán Sancho, Thierry Villemur, Saïd Tazi
CNRS; LAAS; 7 avenue du colonel Roche, F-31077 Toulouse, France
Université de Toulouse; UT1, UT2, UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France
{*sancho,villemur,tazi*}*@laas.fr*

***Abstract*—Collaborative systems provide support for users that work together for achieving a common goal. In the past years, several ad-hoc models have been proposed in order to model collaborative activities in such systems. This paper proposes a shareable model for collaboration, the Generic Collaboration Ontology, that can be used by systems in run-time in order to implement session management and component deployment services. This model is an OWL ontology containing SWRL rules, and therefore it can be processed with standard Semantic Web tools in order to perform inference. This ontology is generic because it does not contain domain-specific knowledge, and it can be extended for specific domains.**

***Keywords*-ontology, collaboration, OWL, SWRL, session, inference**

I. Introduction

Collaborative applications are distributed systems especially designed to provide support to groups of users that act in a coordinated way in order to achieve a common goal. Such applications have been studied since the 1990s in the domain called Computer-Supported Collaborative Work (CSCW). These studies include concepts from very different domains such as Social Sciences, Cognitive Sciences, Human-Machine Interfaces and Distributed Computing in order to maximize the efficiency and ergonomics of CSCW systems.

In the past years, a variety of models and techniques have been developed in the CSCW domain. Applications using these models rely on them in order to represent the possible collaboration schemas and the current system configuration at a given time. These models have been used with more or less success in the implemented systems. However, although many of the modeled elements are common, very often these models are ad-hoc or application-specific, thus limiting their reusability and extensibility. Moreover, each model is described with a different formalism or language, or even worse, they are hard-coded inside the application. This results in a limited interoperability of the systems based in such models. It would be preferable to have common models shared between several applications. These models should be described in standard languages allowing them to be processed with standard tools.

Another disadvantage of existing collaboration models is that they are not well suited for enabling a model-based deployment service. The function of such a service is to deploy (i.e., download, install and configure) the application components necessary on each user device in order to implement the collaboration schema indicated by the model. For example, if, at a given time, the model indicates that an audio connexion must exist between two users, then audio components must be deployed on both users' devices in order to manage that connexion. In old systems, where collaborative software was monolithic, this function was performed statically, and therefore a deployment service was not needed. However, as the systems become more and more dynamic (e.g., in the context of Ubiquitous and Pervasive Computing), deployment needs to be adaptive at run-time, so a dynamic deployment service is needed.

The goal of this paper is to provide a shareable model enabling the development of collaborative applications requiring session management and dynamic component deployment. This model is represented in the OWL ontology language. As far as we know, a common ontology for modeling collaborative sessions has not been proposed yet.

Ontologies have received great attention in the recent years, due to their use for knowledge representation in the Semantic Web domain. The Semantic Web was proposed by Tim Berners-Lee [1] in order to enrich data contained in the World Wide Web. The main idea is to add metadata describing regular Web data (which is only human-readable) in order to make it *understandable* by machines, thus enabling the automation of distributed processing over the Web. Metadata describing the semantics of contents is expressed in several languages such as RDFS (Resource Description Framework Schema, based on RDF) and OWL. OWL (Web Ontology Language) is the Semantic Web standard for describing ontologies [2], which are common vocabularies allowing to model and represent knowledge. The main elements of ontologies are concepts, relations (between two concepts), individuals and axioms. All these elements are based on well-known formalisms such as Description Logics [3] in the case of OWL. Thus, knowledge can be automatically deduced by *inference engines* or *reasoners* (for example, Pellet [4]). These software elements can process an ontology in order to make explicit the implicit knowledge contained in them. Also, rules (expressed in SWRL, the Semantic Web Rule Language [5]) may be included in ontologies and processed by reasoners. Rules add some expressivity to OWL constructs.

For these reasons, OWL seems a good choice for the representation of a shareable collaboration model. Standard tools and frameworks are available and can be used for building and querying model instances. It also enables the sharing of collaboration concepts between several applications. Moreover, the use of reasoning and rules is very useful. For example, they allow deducing, at run-time, the deployment schema that corresponds to a given collaboration configuration.

The contents of this paper are organized as follows. Section II provides an overview of existing collaboration models in CSCW domain in order to analyze the elements to retain in our collaboration ontology. Section III details the elements of the GCO and explains the principles that have guided its design. Section IV presents some guidelines for using an ontology as the core model of a run-time system and provides some examples of systems using the GCO. Finally, Section V concludes and provides some perspectives for future work.

II. Existing collaboration models

This section provides a brief overview of existing models in the CSCW domain and the elements that have to be present in a model for collaborative activities enabling session management and component deployment.

The main models published in the literature are based on set formalisms [6] or first-order logic [7] in order to describe unstructured sessions. In the case of structured sessions, i.e., sessions where relations between members are clearly detailed (e.g., group coordination), models are based on graphs. The modeled elements take part in the definition of group activities. Some of the main elements found in these models are, users, hardware devices, and software tools.

Baudin et al. [8] propose a model capturing the most common elements found in previous systems. The goal of this model is to explicitly represent relationships of information exchange between users in order to keep a tight coupling between communication and network layers. Therefore, this model, which is graph-based, enables the construction of session management services. In this model, a collaborative session is composed of a set of three elements to be managed: users, tools and data flows. Such elements are represented in a unified graph-based model. Vertices represent users, and edges define the relationships between them. An edge going from user U_1 to user U_2 means that U_1 transmits data through a selected tool (e.g., a videoconference tool) to U_2. The type of data and the tools that handle data sent through a flow are defined by edge labels.

The proposed collaboration model is based on data producer/consumer relationships, to represent and process data exchanges for synchronous and interactive work sessions, that. Such sessions handle interactive data flows (e.g. video, real time audio).

This model is simple enough to be easily handled by session designers for various collaborative configurations. Moreover, instances of this model can be automatically taken into account by services or platforms that can be configured by the model. The sessions explicitly designed are managed by model-based platforms.

This model also considers the dynamics of the session: the current session configuration evolves whenever entries or exits of members occur. In the same way, role and function changes of the members already present in the collaborative session introduce modifications of the current graph (for instance a passive user becomes active by making an action and therefore new flows have to be set up). At any time, the current session configuration corresponds to a valid graph.

III. The Generic Collaboration Ontology

This section presents the Generic Collaboration Ontology (GCO)[1]. First, the design principles used for the design of the GCO are presented. Then, the elements present in the GCO are explained in detail.

A. Design Rationale

1) Ontology language: the GCO is expressed in OWL, which is the current web standard for ontology description. Since the expressivity of OWL is not enough for some of the required relations, rules are used. Rules are expressed in SWRL. Standard, open-source tools are available for processing OWL ontologies and SWRL rules.

2) Genericness: the GCO has been designed in order to be as generic as possible. This means that it may be used to model collaboration in any application, regardless of the domain. In this aspect, the GCO can be viewed as an upper ontology that can be extended by domain ontologies in order to model domain-specific concepts and relations. The simplest way of extending this ontology is to use *inhreritance* by defining sub-conceps and sub-relations of the concepts and relations present in the GCO (*is-a* relation).

3) Multi-Layered Architectures: the genericness of the GCO means that it can be used inside a multi-layered architecture. In such case, the GCO may be the core model of the layer that handles collaborative sessions. Domain-specific data may be handled in upper layers, while low-level data, such as network connexions, can be handled in lower layers.

4) Ontology contents: Since the main goal of the GCO is to support collaboration in run-time systems, the concepts and relations present in this ontology have been chosen among those that have been used in collaboration models until today (i.e., those presented in the previous section). For example, it contains concepts representing tools, flows, roles, etc. In order to enable dynamic deployment services based on the GCO, some other elements such as components,

[1] The GCO is a available online at http://homepages.laas.fr/gsancho/ontologies/sessions.owl

nodes hosting devices, etc. have been added to this ontology. The rules associated to the GCO are also designed in order to enable a simpler deployment process by making explicit the deployment schema that must support the collaborative activity described by the ontology.

5) Simplicity: the contents of the GCO have been chosen to enable a complete modeling of collaborative sessions. However, only basic elements have been retained. Therefore, this ontology is lightweight and reasoning and rule processing may be performed at run-time without heavy overhead. Moreover, this simplicity eases the task of designers willing to use or extend this ontology for domain-specific applications.

B. Description of the GCO

The main elements of the Generic Collaboration Ontology are represented in Figure 1. Concepts are represented as round-cornered rectangles, while relations between concepts are represented as arrows going from one concept (the domain of the relation) to another concept (the range of the relation). Individuals are represented as dash-line rectangles.

The basic concept of this ontology is `Node`. A node represents a communicating entity which takes part in a collaborative activity. Nodes may represent human users (i.e. human-controlled software components) but also autonomous software components, agents, etc. The nature of entities is not represented in this generic ontology.

Nodes play a role in the collaborative activity which determines the position of the entity within the collaborative group. Depending on their role, entities will have different functions in the group and they will need to communicate with different group members in order to better achieve the collaboration's goal. This is captured by the concept `Role`. Therefore a relation called `hasRole` links the `Node` and `Role` concepts.

Whether a node is an autonomous software component or it is a human-controlled component, it has to be executed on a physical machine. Such machines are represented by the concept `Device` (`Node` is linked to `Device` by the property `hasHostingDevice`). The execution context of the node will depend on the resources of the device that hosts it. At the present time, a minimal set of device properties is considered, containing IP addresses (`hasIpAddress`), operating system (`hasOS`), available memory (`hasAvailableMemory`), CPU load (`hasAvailableMem`) and battery level (`hasBatteryLevel`). Additional properties could complete this initial list in order to better capture and reason about the execution context.

Entities take part in the collaborative activity by sending and receiving data to/from other entities. The concept `Flow` represents a communication link between two entities. Therefore, `Flow` is linked to `Node` by two properties: `hasSource` and `hasDestination`. In this ontology, flows are considered as being unidirectional, and thus if a bidirectional communication between two nodes is required, it will be represented by two instances of `Flow` with two opposite directions. The `hasSource` property is functional, while `hasDestination` is not functional, i.e., a flow has a single source node, but it may have several destination nodes (thus representing multicast links).

In order to represent the nature of data exchanged through a flow, the `Flow` concept has a functional property called `hasDataType` that relates it to the `DataType` concept. Possible values of data types are captured through the `DataType` individuals `audio`, `text` and `video` (additional data types could be considered). The subconcepts of `Flow` differ in the value of their data type: `AudioFlow`, `TextFlow` and `VideoFlow` (not represented in the figure).

In order to handle data flows, nodes use external software components that are deployed on the same device as them. This enables the separation between business code (implemented in entities' components) and collaboration code (implemented in such external components). These external components are represented by the `Tool` concept. Tools are composed of several components, e.g., a sender component and a receiver component. Therefore the `Tool` concept is related to a concept called `Component` through the property `hasComponent`. Since components handle flows, a property called `managesFlow` links `Component` and `Flow`. Components have a data type (the same as the data type of the flow that they manage) and are deployed on a single device (`isDeployedOn` property which links `Component` and `Device`). The `Component` concept has several subconcepts that represent components depending on the handled data type (`AudioComponent`, `TextComponent` and `VideoComponent`, not represented in the figure) and on the direction of the handled flows (`SenderComponent` and `ReceiverComponent`). `SenderComponent` and `ReceiverComponent` are linked to `Flow` by two sub-relations of `managesFlow`: `sendsFlow` and `receivesFlow`, respectively.

Finally, the `Session` concept represents a set of flows belonging to the same collaborative activity. The `hasFlow` property relates a session to a flow. The inverse property, `belongsToSession`, is functional, i.e., a flow belongs to a single session. Since flows are related to nodes, nodes are indirectly related to one or more sessions depending on the flows that connect them to other entities.

C. Generic collaboration rules

A set of 6 SWRL rules is associated to the GCO in order to express some additional knowledge and to enable deployment-related inference. This section provides a description of the main rules associated to the GCO.

Let us consider the first rule:

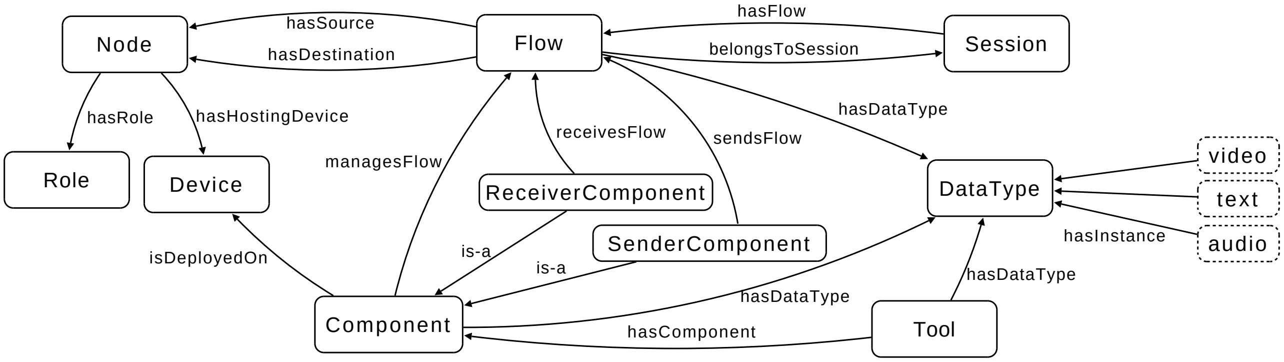


Figure 1. Main elements of the Generic Collaboration Ontology.

```
AudioFlow(?af) => hasDataType(?f,audio)
```

This rule allows expressing that the data type of `AudioFlows` is `audio`. Similar rules exist for video an text flows. The second rule is:

```
Flow(?f) & belongsToSession(?f,?s)
& hasDataType(?f,?dt)
& hasSource(?f,?src)
& hasDestination(?f,?dst)
& swrlx:createOWLThing(?sc,?src,?s)
& swrlx:createOWLThing(?rc,?dst,?f)
=> SenderComponent(?sc)
& hasDataType(?sc,?dt)
& isDeployedOn(?sc,?src)
& sendsFlow(?sc,?f)
& ReceiverComponent(?rc)
& hasDataType(?rc,?dt)
& isDeployedOn(?rc,?dst)
& receivesFlow(?rc,?f)
```

This rule states that, whenever a flow belonging to a session is found between two nodes, a `SenderComponent` has to be present in the source node and a `ReceiverComponent` has to be present on the destination node. These components send and receive, respectively, the flow, and they have the same data type as the flow. This rule uses the SWRL built-in `createOWLThing` that allows creating new individuals. Please note that the first `createOWLThing` matches the source node and the session, while the second matches the destination node and the flow. This choice enables multicast flows where a single sender component sends several flows to several receiver components.

These rules are generic w.r.t. collaboration, because they do not depend on the particular domain of the collaborative application.

IV. Use of the GCO in Runtime Systems

As explained before, the GCO has been designed to be used at run-time as the core model of systems providing support for collaborative activities. This section details this use and gives some examples of collaborative systems that use the GCO.

A. *Using ontologies in run-time systems*

An ontology may be considered as a meta-model which describes the possible concepts and relations of a given domain. Actual instances of this meta-model are represented by individuals of the concepts available in the ontology. Such individuals (and the relations between them) may be used in order to represent the state of the application at a given time. Relations and concepts are fixed at design-time, while individuals representing the state are created at run-time. In order to use an ontology as the core model in a run-time system, the system must be able to perform the following tasks:

- read the concepts and relations existing in the ontology;
- read/modify the individuals existing in the ontology and the values of their properties;
- create new individuals and set the values of properties;
- perform reasoning and rule processing over the ontology and its individuals.

The tools made available in the context of the Semantic Web enable the execution of these tasks. Implementations of APIs like OWL API [9] or Protégé-OWL API [10] allow performing the four first tasks programmatically. Reasoning can be performed by OWL reasoners such as Pellet [4]. Most reasoners are also capable of processing SWRL rules; however, SWRL *built-ins* are not fully supported yet. Therefore, it may be necessary to use rule engines such as Jess[2] in order to process rules containing such built-ins. Both reasoners and rule engines can be executed programmatically in order to process in-memory OWL models.

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

The presented tools enable the creation of programs that modify the individuals of an ontology in order to represent the current state of of the system at every time. However, if reasoning and rules are used to deduce knowledge from individuals, the monotonic nature of OWL inference may represent a problem. Indeed, OWL does not support non-monotonic inference [2], [11]. This means that reasoning and rules can not modify (addition or removal) the information contained in an ontology. They only allow finding implicit knowledge contained in the ontology and making it explicit. For example, if the processing of a rule in the GCO results in the creation of an individual of the class `Flow` whose source is $node_A$ and whose destination is $node_B$, this information will always remain in the ontology. No other rule can remove it afterwards. If the application needs to remove this individual in order to reflect a new state, it can do it programmatically, but it can be very tricky and unpractical (or even impossible) to keep a track of which information has been inferred and to decide what has to be deleted at every moment.

The solution to this problem is to use the inference capabilities of OWL in a capture-inference-results loop such as the one depicted in Figure 2. The first step if to capture the *state of the world* that is modeled by the ontology. This is done by the code of the application using the ontology. Then, this state is introduced in the ontology by creating individuals of the available concepts and by establishing relations between these individuals through object properties. The result is a set of ontology individuals related between them reflecting the state of the modeled world. For example, if a system using the GCO has, at a given time, three users connected, there will be three individuals of the class `Node` representing these users, each one related to one individual of the class `Role` representing the role of the user in the group. Once this model has been built, the resulting ontology can be processed by the inference and rule engines. The result of this step is a new version of the ontology where new individuals and relations may have been introduced. In the example of the GCO, this step results, e.g., in the creation of several individuals of the class `Flow` having the existing `Node` individuals as values for the properties `hasSource` and `hasDestination`. This new model contains the new state of the world that has to be achieved. Therefore, the application code can read this model and perform the actions necessary in order to achieve this state. In the GCO example, the new `Flow` instances will be found and therefore the application will effectively set up this new flows between the users' devices. Whenever the state of the world is changed (e.g., when one of the users leaves), the whole loop has to be repeated in order adapt the response of the application to the new state.

The presented loop is discrete; the results of a step are valid until the next change in the state of the world. Whenever a change occurs, the whole loop is executed again in order to get the new results. Because of the monotonicity of OWL inference, the new state can not be represented by directly modifying the resulting ontology individuals; it would be necessary to delete all the inferred knowledge. Otherwise, the next inference process will result in an incoherent (inconsistent) ontology.

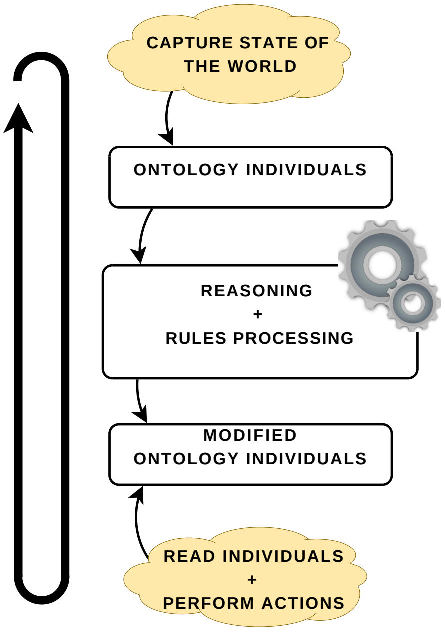

CAPTURE STATE OF THE WORLD
ONTOLOGY INDIVIDUALS
REASONING + RULES PROCESSING
MODIFIED ONTOLOGY INDIVIDUALS
READ INDIVIDUALS + PERFORM ACTIONS

Figure 2. Capture-inference-results loop for run-time systems using ontology reasoning.

B. Systems using the GCO

In a recent work [12], [13] we have proposed a modeling approach and a framework enabling the design and implementation of collaborative applications for ubiquitous computing environments. Ubiquitous Computing provides a new range of challenges and opportunities for collaborative applications. Indeed, the fact that mobile users carrying smart devices are immersed in intelligent environments and the availability of contextual data may greatly enhance the possibilities of collaborative applications. The presented framework uses a multi-layered approach for enabling applications that can be adapted to both high-level requirements and low-level constraints. The proposed layers are called application layer, collaboration layer and middleware layer. The core model of the application layer is the GCO. The models of the application layer are domain-specific ontologies and rules that extend the GCO in order to capture the specific collaboration knowledge of the considered domains. This ontology is processed in a capture-inference-results loop as explained in the previous section. The results of this process are translated into a graph model, which is the core model of the middleware layer, and then it is processed with graph-transformation techniques. This allows taking into account high-level requirements at the application and collaboration layers, while low-level requirements are handled at the middleware layer. The resulting graph is a low-level, detailed deployment descriptor that is used by the deployment service in order to carry out the deployment of components needed for supporting collaborative activities.

Bouassida Rodriguez et al. [14] propose an Emergency Response and Crisis Management System (ERCMS) that uses the GCO as the core collaboration model. ERCMSs support collaboration of policemen, firemen and physicians in order to better handle critical situations such as fires, earthquakes, terrorist attacks, etc. The proposed system is adaptive and takes into account the evolution of communication and processing resources in order to guarantee the required QoS properties. Non-functional properties are modeled in an OWL ontology that extends the GCO by relating `QualityAttributes` to the `Component` and `Device` concepts of the GCO. A domain-specific OWL ontology is used in order to describe ERCMS-specific collaborative knowledge. This ontology extends the GCO by providing sub-concepts of the `Flow` and `Node` concepts of the GCO. Several SWRL rules are provided for implementing adaptation transformations that handle context changes. However, the authors do not explain how their system uses the proposed ontologies and rules at run-time, neither they explain how the problem of the monotonic nature of OWL inference is handled.

V. Conclusion and Future Work

This paper has presented the GCO, a generic collaboration ontology that represents knowledge about session-oriented collaboration. This ontology is generic because it can be extended in order to model domain-specific collaboration knowledge. Rules associated to the GCO allow implementing ontology-driven systems using the GCO as their core collaboration model for implementing session management and deployment services. Explanations of how this usage of the GCO in run-time systems have also been provided.

Perspectives for future work include designing domain-specific ontologies that extend the GCO for several domains and building systems that use the GCO as their model for collaboration activities. The framework described in Section IV-B is currently being implemented, as well as proof-of-concept applications that use it for modeling and implementing collaborative activities.

Acknowledgment

The authors would like to thank ITEA2 USENET project for supporting this work.

References

[1] T. Berners-Lee, “A Roadmap to the Semantic Web,” W3C, Sep. 1998, http://www.w3.org/DesignIssues/Semantic, last access date: July 2010.

[2] M. K. Smith, C. Welty, and D. L. McGuinness, “OWL Web Ontology Language Guide,” W3C Recommendation, Feb. 2004, http://www.w3.org/TR/owl-guide/, last access date: July 2010.

[3] D. F. Baader, D. Calvanese, D. L. McGuinness, P. Patel-Schneider, and D. Nardi, *The Description Logic Handbook. Theory, Implementation, and Applications*.

[4] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, “Pellet: A practical OWL-DL reasoner,” *Web Semant.*, vol. 5, no. 2, pp. 51–53, 2007.

[5] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, “SWRL: A Semantic Web Rule Language Combining OWL and RuleML,” W3C Member Submission 21 May 2004, 2004.

[6] W. K. Edwards, “Session management for collaborative applications,” in *CSCW ’94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*. New York, NY, USA: ACM, 1994, pp. 323–330.

[7] M. Rusinkiewicz, W. Klas, T. Tesch, J. Wäsch, and P. Muth, “Towards a Cooperative Transaction Model - The Cooperative Activity Model,” in *Proceedings of the 21th International Conference on Very Large Data Bases*. San Francisco, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 194–205.

[8] V. Baudin, K. Drira, T. Villemur, and S. Tazi, “A model-driven approach for synchronous dynamic collaborative e-learning,” in *E-Education applications: human factors and innovative approaches*. Ed. C. Ghaoui, Information Science Publishing, ISBN 1-59140-292-1, 2004, pp. 44–65.

[9] M. Horridge and S. Bechhofer, “The OWL API: A Java API for Working with OWL 2 Ontologies,” in *OWLED*, ser. CEUR Workshop Proceedings, R. Hoekstra and P. F. Patel-Schneider, Eds., vol. 529. CEUR-WS.org, 2008.

[10] H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen, “The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications,” *The Semantic Web - ISWC 2004*, pp. 229–243, 2004.

[11] J. McCarthy, “Generality in artificial intelligence,” *Commun. ACM*, vol. 30, no. 12, pp. 1030–1035, 1987.

[12] I. Bouassida Rodriguez, G. Sancho, T. Villemur, S. Tazi, and K. Drira, “A model-driven adaptive approach for collaborative ubiquitous systems,” in *AUPC 09: Proceedings of the 3rd workshop on Agent-oriented software engineering challenges for ubiquitous and pervasive computing*. London, United Kingdom: ACM, 2009, pp. 15–20.

[13] G. Sancho, I. Bouassida, T.Villemur, S. Tazi, and K. Drira, “A model-driven adaptive framework for collaborative ubiquitous systems,” in *Proceedings of the 9th Annual International Conference on New Technologies of Distributed Systems, NOTERE 2009*, Montreal (Canada), July 2009, pp. 233–244.

[14] I. Bouassida, J. Lacouture, and K. Drira, “Semantic Driven Self-Adaptation of Communications Applied to ERCMS,” in *The 24th IEEE International Conference on Advanced Information Networking and Applications*, Perth (Australia), Apr. 2010.

A Document Authoring System for Credible Enterprise Reporting with Data Analysis from Data Warehouse

Masao Mori
IR Office, Kyushu University,
Fukuoka, Japan
mori@ir.kyushu-u.ac.jp

Toshie Tanaka
IR Office, Kyushu University,
Fukuoka, Japan
tanaka@ir.kyushu-u.ac.jp

Sachio Hirokawa
Research Institute for Information Technology,
Kyushu University, Fukuoka, Japan
hirokawa@cc.kyushu-u.ac.jp

***Abstract*—In rapid progress of information technology, we are facing difficulties, "information explosion". From standpoint of using enormous quantity of data, there are many researches such as information retrieval and clustering information. On the other hand, in terms of creating credible enterprise reports, information explosion also becomes a big problem. If most of digital documents are unstructured, report writers may have significant difficulties with management and arrangement of digital documents. Actually in the case of university evaluations, report writers have been confronted with that difficulties. In addition, quantitative data from data warehouse is indispensable for enterprise reports. In this paper, we developed a document authoring system cooperating with data warehouse to settle these problems from viewpoint of reusing and reconstructing components of reports.**

***Keywords*-digital document; data warehouse; accreditation; knowledge management; web service**

I. Introduction

In recent years opportunities of enterprise reporting in companies, institutions and universities have been increasing rapidly. So that business intelligence and content management system for enterprise reporting are desirable, for instance, Priebe[1]. What is required to create credible enterprise reports? Morimoto et al.[2] asserts the following four processes of enterprise reporting from viewpoint of knowledge management: (1) collecting and accumulating documents, (2) searching and browsing documents, (3) extracting and identifying documents and (4) creating credible reports. In order to realize these processes completely, information must be structured. DITA[3] is one of the ideal architectures to extract information from documents effectively and to manage documents efficiently. However, not infrequently, non-structured information exceeds structured information, especially on-the-spot of university evaluations.

All Japaneses universities are obliged to be evaluated by certified organization, called *institutional certified evaluation and accreditation*. In addition, all Japaneses national universities must be evaluated for the purpose of information disclosure to government and nation, called *national university corporation evaluation*. They are called *university evaluations* which is undergone every six years. Universities must prepare self-assessment reports for university evaluations. Educational and research activities of university vary in many ways. In Kyushu university, one of national universities in Japan, though documents of committee and faculty council were stored, they had not yet been managed systematically. How to reuse these documents becomes a big problem. Authors of this paper have been supporting faculties and bureaus to create university evaluation reports. As in Figure1, the amount of document in evaluation report of Kyushu university 2009 was so large-scaled that it was hard even to fix formats of documents. In addition, many items and themes appear many times in both reports. So the writer must be thoughtful for consistency of both reports.

Fields	Schools	Contents of report		
		Sections	Viewpoints	Pages
Education	31	8	12	959
Research	20	5	5	311

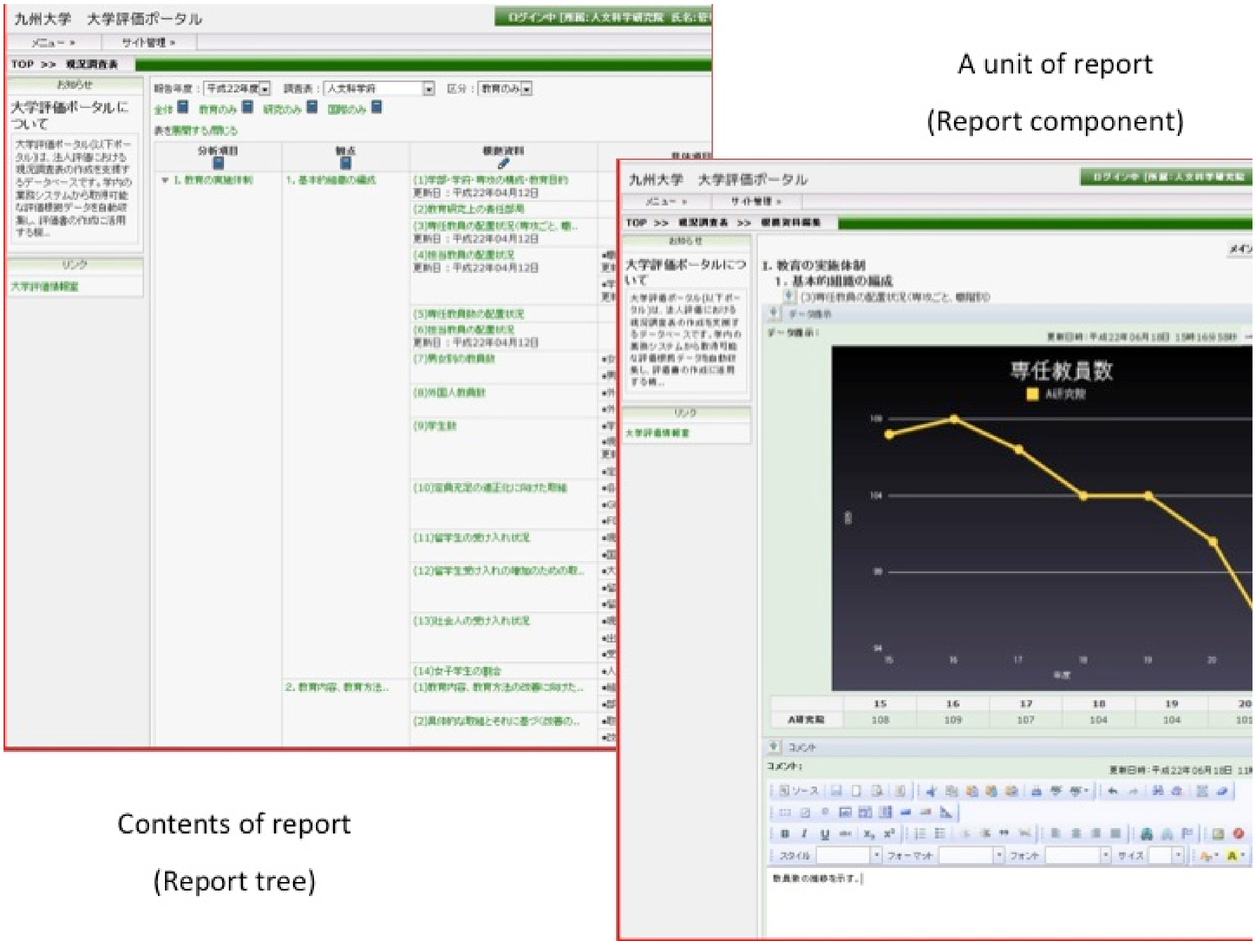
Figure 1. An example of amounts of documents in the corporation evaluation report of educational and research activity of Kyushu university 2009

From our experience to support creation of evaluation reports, we have developed a document authoring system for enterprise report, especially for university evaluations, cooperating with data warehouse. In order to manage unstructured information efficiently, the proposing system provides users with a simple and uniform data structure for report components. Users can create enterprise reports by arranging report components in the tree structure of sections. Moreover, by reusing report components users can make sure of consistency of enterprise report.

Our system challenges the two targets as follows: (1) management of items and themes which appear frequently in various enterprise reports, (2) light-weight cooperation with data warehouse. This system is developed using Ruby on Rails and MySQL. Demonstration of our system can be seen on Youtube[1].

The paper is structured as follows: In Section 2 we review related works. In Section 3 we overview our system and introduce three main concepts, report components, report

[1] `http://www.youtube.com/watch?v=okAT6aseks8`



	15	16	17	18	19	20
A研究院	108	109	107	104	104	101

Figure 2. Views of the document authoring system

tree and data analysis queries. In Section 4 we present features of our approach comparing with related work. We conclude the paper with summary and future work.

II. Related work

We start to discuss related work by reviewing the assertion of processes for enterprise report in Morimoto et al.[2]: (1) collecting and accumulating documents, (2) searching and browsing documents, (3) extracting and identifying documents and (4) creating credible reports.

Considering document management, information retrieval is indispensable for accumulated digital documents. Beyer et al.[4] propose a method to discover patterns and rules of texts in structured documents in order to generate efficient search index. Linked Data[5] would be helpful to capture relationship between digital documents if they were structured. But we found that the prime consideration for documents in enterprise reporting, especially in university evaluation, is meta-data of digital documents and materials, such as their jurisdiction, creators and meanings of the documents. As a university is a complex organization consisting of many departments and bureaus with autonomy, meta-data of documents is indispensable for document management in the scene of university evaluations.

DITA[3][6] is a document architecture for extraction and management of documents. DITA enables users to extract and update information efficiently in large amounts of documents[7]. In order to adopt DITA and Linked Data, it is required to define an ontology for knowledge of enterprise. Since it is difficult to apply an ontology to present progressive enterprise processes and legacy systems, we decide to extract text from digital document by hand and to collect minimum concrete information (such as “Section 2 on page 23”) as meta-data about digital documents.

Generally speaking, accumulating daily reports ensures enterprise reports, moreover it is advisable to study how to obtain meanings and attributes of documents[2]. If an enterprise report is required to be prompt, integration of document creation with OLAP is desirable[1]. In the case of university evaluation reports, frequency of reports is much lower than daily reports in companies. Actually evaluation report is usually conducted every year or every month at most. A long-term vision rather than promptness is necessary for university management. One of important requests in university evaluations is to select documents efficiently and to organize them effectively rather than automatic reporting function. The proposing system provides users with an interactive interface to select documents and organize reports.

Integration of structured data in data warehouse and unstructured data in texts on news sites and blogs has been studied in [1][5][8][9]. Most of them are based on information retrieval and assume that ontology for structured data is given, whereas we assume that ontology is not given but the design of enterprise reports is given, like university evaluations. Our approach is different from those related work in terms of these assumptions.

III. Overview

A. Report Component and Report Tree

In this subsection, we will introduce the document authoring system for enterprise reporting (DASER for short). As we mentioned in the introduction, it is important to provide users with a uniform data structure in order to bundle essential information of materials and documents. A data structure, *report component*, is a unit in DASER, which consists of seven elements as follows:

1) id,
2) title (user input),
3) comment (user input),
4) data analysis query (user input),
5) data analysis,
6) attached documents (user input), and
7) meta-data.

Users may input data into attributes such as comment, data analysis query and attached documents. *Data analysis* is visualization of data obtained from data warehouse through "*data analysis query*" (DAQ for short). DAQ is URL of a CGI program in data warehouse. We will discuss DAQ in the next section. Meta-data is owner information and timestamp. Each report component have visualizing function for CSV data obtained from DAQ.

The window on the right in Figure 2 is an example of report component. The graph is generated from CSV data which is obtained from data warehouse through DAQ. Note that the visualizing function does not depend on DAQ. One can visualize static CSV files located in other web server.

In DASER, we can define structure of enterprise report by giving a tree structure with report components as leaves and sections as internal nodes. This is called a *report tree*. Report tree is changeable corresponding to contents of every enterprise report, and it also can be changed depending on individual needs from users. Report tree can be construct with report component as leaf nodes, and with the root node and internal nodes. A root node and internal nodes have the same data structure as a report component and additional attributes as follows:

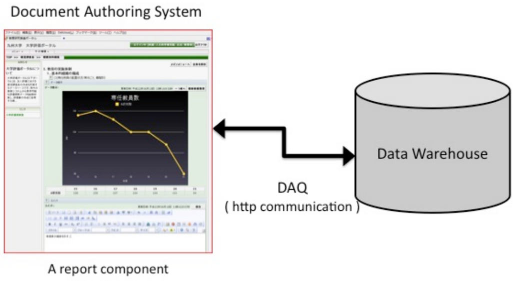


Figure 3. The report authoring system and data warehouse

8) a list of ids of children and
9) its parent's id.

Note that each report component does not depend on the definition of report tree.

B. Data Analysis Query

In this section we introduce data warehouse (DW for short) and its data analysis query. DAQ is WebAPI of DW. Data sources of DW is backup data of operational (business) systems. In the context of university evaluation, for example, they are information about students, teachers, teaching, research and finance of university. Flat files such as spreadsheets, are also data source of DW.

Administrator of DW provides users with programs in order to analyze data in DW. That is called *data analysis query*. As DAQs are implemented as CGI programs, one can access DW with DAQs over restful HTTP communication. DAQs return data in CSV format.

Let us consider the case of analyzing international students enrollment. One need to calculate numbers of students for every year and every department in order to show their changes. For example, the DAQ

```
http://dw.mydom/int_stdt.cgi?yr=5&dpt=eng
```

returns CSV data about changes of international students in the department of engineering (`dpt=eng`) for the past five years (`yr=5`). This DAQ is available for other departments and other year terms by changing parameters.

IV. Features

In the introduction, we mentioned that our challenges are: (1) management of items and themes which appear frequently in various enterprise reports, and (2) light-weight cooperation with data warehouse. In this section we will see achievement to the challenges.

A. Consistency and Credibility

Firstly we discuss how the proposing system contributes to consistency for enterprise reports.

Generally speaking, contents of an enterprise report form a tree structure. Leaf nodes are topics and themes and internal nodes are sections and chapters. So we define a report component as a leaf node, which is a data structure with seven attributes, and chapters and section as internal nodes. When users create multiple reports in such as our case of two university evaluations, what user have to do is setting each report tree corresponding to a configuration of each report. Then DASER flexibly generates multiple reports. Even if some report components appear many times in different reports, DASER ensures consistency and credibility between different reports. Related work, such as [1][5][8][9], have not focused on the problem of multiple reports. This is one of unique features of our approach.

B. Light-weight cooperation and its effectiveness

DASER is connected to DW only through DAQs by restful http communication which is one of web service techniques. We could successfully develop DASER and DW separately. In other words, DW can offer the CSV data to other service besides DASER, and DASER can refer to static CSV files from other data source besides DW.

Sharing data warehouse inside of intranet has been a trend for a decade [10]. Our approach is to develop an integration of qualitative data and quantitative data for enterprise reporting, whereas we must develop data warehouse for not only reporting but also sharing information inside of our university. This situation is different from [8][1].

C. Flood of unstructured and valuable XML data

In order to accomplish information disclosure, enterprise documents are always accumulated. This issue is for not only big organizations such as big universities, for but also any small organizations such as elementary schools.

Unfortunately, in many universities and schools in Japan, most of their digital documents, like word processor files and spread sheets, are unstructured data. That is why we must assume nonexistence of ontology for our approach. When user creates an enterprise report on our system, she/he is supposed to set up report components and report trees. Giving report components and report trees would lead to the ontology for the enterprise report. That is one of unique feature of our system.

V. Conclusion and future work

In this paper we developed a document authoring system for enterprise reporting cooperating with data warehouse. And we realized a light-weight cooperation between our system by using the technique of restful http communication.

Two problems still remain. First problem is flexibility of report component. Under current configuration of DASER, user cannot variously set the contents of report component to the context of each enterprise reporting. Second problem is flexibility of composing results of DAQ. Cross tabulation of two or more results of DAQs is impossible. From our researches like [11][12], it is considerable to apply the method of web mash-ups to the second problem.

References

[1] T. Priebe and G. Pernul, “Ontology-based integration of olap and information retrieval,” in *Proceedings of the 14th International Workshop on Database and Expert Systems Applications*, 2003, p. 610.

[2] Y. Morimoto, H. Mase, and H. Tsuji, “Perspectives on reuse process support systems for document-type knowledge,” in *Human-Computer Interaction, Part IV, HCII 2007*, ser. Lecture Note in Computer Science 4553, 2007, pp. 682–691.

[3] O. Standard, *DITA Version 1.1 Architectural Specification*, OASIS, 2007.

[4] K. Beyer, V. Ercegovac, and R. K. et al., “Towards a scalable enterprise content analytics platform,” *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 2009.

[5] C. Bizer, T. Health, K. Idehen, and T. Berners-Lee, “Linked data on the web (ldow2008),” in *Proceeding of the 17th international conference on World Wide Web*, 2008, pp. 1265–1266.

[6] M. Priestley, “Dita xml: A reuse by reference architecture for technical documentation,” in *Proceedings of SIGDOC’01*, October 2001, pp. 152–156, santa Fe, New Mexico, USA.

[7] O. Dìaz, F. I. Anfurrutia, and J. Kortabitarte, “Using dita for documenting software product lines,” in *Proceedings of the 9th ACM symposium on Document engineering*. Association for Computing Machinery, 2009, pp. 231–240.

[8] A. Ferrández and J. Peral, “The benefits of the interaction between data warehouses and question answering,” in *Proceedings of the 2010 EDBT/ICDT Workshops*, ser. ACM International Conference Proceeding Series, vol. 426, 2010.

[9] B. Riger, A. Kleber, and E. von Maur, “Metadatabased integration of qualitative and quantitative information resources approaching knowledge management,” in *ECIS 2000 Proceedings*, 2000.

[10] R. Kimball and R. Merz, *The Data Webhouse Toolkit: Building the Web-Enabled Data Warehouse*. Wiley, 2000.

[11] M. Mori, T. Nakatoh, and S. Hirokawa, “Links anc cycles of web databases,” in *The 4th Italian Workshop on Semantic Web Applications and Perspectives*, 2007, pp. 21–30.

[12] ——, “Functional composition of web databases,” in *Proceedings of International Conference Asian Digital Libraries 2006*, ser. Lecture Note in Computer Science 4312. Springer Verlag, 2006, pp. 439–448.

[13] S. Mushhad, M. Gilani, J. Ahmed, and M. A. Abbas, “Electronic document management: A paperless university model,” in *Proceedings of 2009 2nd IEEE International Conference on Computer Science and Information Technology*, 2009, pp. 440–444.