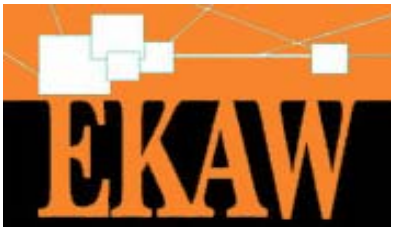


EKAW 2008
16th International Conference on
Knowledge Engineering and Knowledge Management
Knowledge Patterns

Poster and Demo Proceedings

Poster and Demo Proceedings - Content

The Cell Cycle Ontology: a step towards Semantic Systems Biology

Poster paper

Erick Antezana[1,2], Mikel Egaña[3], Ward Blondé[1,2], Vladimir Mironov[4], Robert Stevens[3], Bernard De Baets[5], and Martin Kuiper[1,4]

[1] Dept. of Plant Systems Biology, VIB, Gent, Belgium
[2] Dept. of Molecular Genetics, Ghent University, Gent, Belgium
[3] School of Computer Science, University of Manchester, UK
[4] Dept. of Biology, Norwegian University of Science and Technology, Norway
[5] Dept. of Mathematics, Biometrics and Process Control, Ghent University, Belgium
{erant|wablo|vlmir|makui}@psb.ugent.be
{eganaarm|stevensr}@cs.man.ac.uk
bdebaets@ugent.be

Abstract. The terms and relationships provided by existing bio-ontologies only represent a limited set of features of biological regulatory processes. As current bio-ontologies only explicitly capture a small part of our biological understanding, the potential of applying computational analysis on such knowledge remains limited. The Cell Cycle Ontology (CCO) is designed to capture detailed knowledge of the cell cycle process by combining representations from several sources. CCO is an application ontology that is supplied as an integrated turnkey system for exploratory analysis, advanced querying, and automated reasoning. Linking and converting bio-ontologies to semantic web languages, such as OWL, opens possibilities to widely exploit computational approaches for knowledge visualization, retrieval and automated inference which in turn can support systems biology approaches.

1 Rationale

Findings in life science are being reported at an increasingly rapid rate. Such information finds its way in diverse locations and its integration into a common *format* is recognized as a critical step toward hypothesis building [1] and exploitation by researchers and automated applications [2]. To obtain a powerful structuring and synthesis of all available biological knowledge it is essential to build an efficient information retrieval and management system. Such a system requires an extensive combination of data extraction methods, data format conversions and a variety of information sources. Biological knowledge integration is recognized as a critical knowledge gap in science [3] and deemed essential for the future of the biosciences since dissemination and exploitation of the knowledge by automated applications will provide critical assistance to researchers who need to access and connect the diverse information sources.

2 Current status

We developed the Cell Cycle Ontology (CCO), an application ontology [4], to cover the domain of cell cycle research [5]. CCO supports 4 organisms: Human, Arabidopsis, Baker's yeast and Fission yeast with separate ontologies but also one integrated ontology. CCO holds more that 65000 concepts (more than 52000 bio-molecules and over 9000 interactions) and more than 20 types of relationships. A set of PERL modules [6] has been developed to deal with format conversions (e.g. OBO to OWL) and in particular to manipulate ontologies (in tasks such as getting sub-ontologies and merging them). CCO comprises data from a number of resources such as Gene Ontology (GO) [7], Relations Ontology (RO) [8], IntAct (MI) [9], NCBI taxonomy [10], UniProt [11] as well as orthology data. An automatic pipeline builds CCO from scratch on a monthly basis: during the first phase some existing ontologies (GO, RO, MI, in-house) are automatically retrieved, integrated and merged, producing in turn a core cell cycle ontology. Then, organism-specific protein and gene data are added from UniProt and from the GOA files [12], generating 4 organism-specific ontologies. Those 4 ontologies are merged and more terms are included from an ontology built from the OrthoMCL[6] execution on the cell cycle proteins. Finally, during the maintenance phase, a semantic improvement on the OWL version is performed: ontology design patterns [13] are included using the Ontology Pre-Processor Language [14]. CCO term identifiers are consistently and systematically handled. These identifiers have the form: **CCO:Xnnnnnnn**, where **CCO** denotes the ontology namespace, **X** the subnamespace (such as **G** for **gene**) and **nnnnnnn** denotes a unique number. The resulting CCO is designed to provide a richer view of the cell cycle regulatory process, in particular by accommodating the intrinsic dynamics of this process. CCO is available in different formats, there is also a SPARQL endpoint[7] for exploiting the RDF export. Visual exploration can be done via the BioPortal[8], OLS[9] or the Ontology Online service[10].

3 Outlook into the future

Integration of many more data sources is foreseen (e.g. miRNA data). CCO provides a test bed for the deployment of advanced reasoning approaches for knowledge discovery and hypotheses generation. Immediate CCO developments include research on reasoning at different levels of granularity after integrating non-crisp data and weighting the current evidence of the existing biological relations. An important extension of the reasoning capability is required to deal with *fuzziness*, a component that is usually present in biological data. In that sense, a combination of both issues (granularity and fuzzy data) will be considered.

[6] http://www.orthomcl.org
[7] http://www.cellcycleontology.org/query/sparql
[8] http://bioportal.bioontology.org/
[9] http://www.ebi.ac.uk/ontology-lookup/
[10] http://ontologyonline.org/visualisation/c/CellCycleOntology/biological+entity

Acknowledgements

This work was funded by the EU FP6 (LSHG-CT-2004-512143). EA was funded by the European Science Foundation (ESF) for the activity entitled Frontiers of Functional Genomics, ME by the University of Manchester and the EPSRC.

References

1. Pennisi, E.: How will big pictures emerge from a sea of biological data? Science **309**(5731) (July 2005) 94
2. Gardner, S.P.: Ontologies and semantic data integration. Drug Discovery Today **10**(14) (July 2005) 1001–1007
3. Cannata, N., Merelli, E., Altman, R.B.: Time to organize the bioinformatics resourceome. PLoS Comput Biol **1**(7) (Dec 2005) e76
4. van Heijst, G., Schreiber, A.T., Wielinga, B.J.: Using explicit ontologies in kbs development. Int. J. Hum.-Comput. Stud. **46**(2-3) (1997) 183–292
5. Antezana, E., Tsiporkova, E., Mironov, V., Kuiper, M.: A cell-cycle knowledge integration framework. In Leser, U., Naumann, F., Eckman, B.A., eds.: DILS. Volume 4075 of Lecture Notes in Computer Science., Springer (2006) 19–34
6. Antezana, E., Egaña, M., Baets, B.D., Kuiper, M., Mironov, V.: Onto-perl. Bioinformatics **24**(6) (2008) 885–887
7. Gene Ontology Consortium: Gene Ontology: tool for the unification of biology. Nat Genet **23**(May) (2000) 25–29
8. Smith, B., Ceusters, W., Klagges, B., Kohler, J., Kumar, A., Lomax, J., Mungall, C., Neuhaus, F., Rector, A., Rosse, C.: Relations in Biomedical Ontologies. Genome Biology **6** (2005) R46
9. Hermjakob, H., Montecchi-Palazzi, L., Lewington, C., Mudali, S., Kerrien, S., Orchard, S.E., Vingron, M., Roechert, B., Roepstorff, P., Valencia, A., Margalit, H., Armstrong, J., Bairoch, A., Cesareni, G., Sherman, D.J., Apweiler, R.: Intact: an open source molecular interaction database. Nucleic Acids Research **32**(Database-Issue) (2004) 452–455
10. Wheeler, D., Chappey, C., Lash, A., Leipe, D., Madden, T., Schuler, G., Tatusova, T., Rapp, B.: Database resources of the national center for biotechnology information. Nucleic Acids Research **28**(1) (2000) 10–14
11. The UniProt Consortium: The universal protein resource (uniprot). Nucleic Acids Research **36**(Database-Issue) (2008) 190–195
12. Camon, E., Magrane, M., Barrell, D., Lee, V., Dimmer, E., Maslen, J., Binns, D., Harte, N., Lopez, R., Apweiler, R.: The gene ontology annotation (goa) database: sharing knowledge in uniprot with gene ontology. Nucleic Acids Research **32** (2004) D262
13. Egaña, M., Antezana, E., Kuiper, M., Stevens, R.: Ontology Design Patterns for bio-ontologies: a case study on the Cell Cycle Ontology. BMC bioinformatics **9(Suppl 5)** (2008) S1
14. Egaña, M., Antezana, E., Stevens, R.: Transforming the Axiomisation of Ontologies: The Ontology Pre-Processor Language. OWLED 2008

Supporting Subject Experts with Ontology Maintenance

Claudio Baldassarre

Knowledge Media Institute (KMi), The Open University
c.baldassarre@open.ac.uk

Abstract. Semantic technologies are an emblematic example of technological shift particularly interesting in the field of knowledge management. We specifically look in the scope of knowledge maintenance at the problem of how hard can be to re-address knowledge workers to maintain different knowledge model types. In this paper we describe the vision of a knowledge maintenance framework whose objective is to set up a network of maintenance spaces where to re-render knowledge modifications according to the formality of the local knowledge model. We illustrate an example of applicability in the case of AGROVOC thesaurus.

1 Introduction

Existing studies like in [10] pointed out how shifting to new technologies, as in the case of semantic for knowledge maintenance equally impacts on legacy systems and knowledge workers.

Literature offers rather sharp separation between two approaches to reduce shifting effects, and supporting the maintenance of semantic knowledge models. Examples from [1], [7],[4], and [2] can be, however, seen as *component technologies* contributing to the knowledge maintenance. They are usually pluggable into larger frameworks and all offer to help the workers in partial maintenance tasks. Alternatively, fully fledged (ii)*integrated solutions* like [3],[5],[9],and [8] were proposed to adopt a common – often newly designed – maintenance paradigm to increase workers' cognitive fit with the system.

We address the limitations of requiring substantial skills of ontology modeling to parse output from the referenced applications, or similarly the need, for some users, to undergo further training to approach a new working paradigm. We hence work to improve the seamless handover between traditional and new maintenance technologies, setting up a network of maintenance spaces capable to communicate about the changes in knowledge on the levels suiting the formality of their models.

2 Knowledge maintenance via knowledge re-purposing

To achieve the research objective we interpret each form of knowledge maintenance as encapsulated; i.e., we define **maintenance space** as characterized

by *one knowledge model*, *one set of worker's expertise*, and *one work practice paradigm* (e.g., DBMS, CMS, OMS, etc.) To improve on the existing solutions we research the requirements and practices adopted in the model maintenance, finally we intend to encode maintained knowledge into ontologies, and design a set of maintenance patterns.

When a maintenance task is performed in one of the spaces (i.e., the source space), a procedure is triggered to search and select which design patterns can be instantiated. The user is involved in describing current task with more natural terms, and these descriptions are reused in another space (i.e., target space). The description generated in this way will contain information about: subject of change, interpretation for target space, and reference to the pattern(s) instantiated – all wrapped into a "maintenance message".

When the target space gets the message, another negotiation procedure, (taking the message as input), will drive the maintainer to render the change equivalently in the target space. S/he will adopt local procedures to amend the local knowledge model. We envision this is a foundation of a new type of maintenance – **Knowledge Re-purposing**.

3 A case study: AGROVOC thesaurus

Currently, knowledge workers in FAO are working on an ontology of AGROVOC [6], a non-trivial task depending on the capacity to break down agricultural scientific knowledge from lexical properties of the terms. Whether the ontology will become mainstream or not, AGROVOC thesaurus will remain. FAO will still need its expert maintainers conceptualizing the domain in terms of Broader Term (BT), Narrower Term (NT), and Related Term (RT) relationships, and thus maintaining this large thesaurus. Needless to say that those relations are much more specialized in to the equivalent ontological model(s). This finer level of detail is also the reason why it is necessary to have specialists in handling both information formats.

Suppose that an AGROVOC maintainer adds the following relation:

Acid Soil **NT** Chemical Soil Types

While this is a coherent statement within its scope, for ontology maintainers is non-trivial modelling choice if a more specific Chemical Soil Type (RDFS:SUBCLASSOF), or a specific instance of Chemical Soil Type (RDF:TYPE) is intended. Hence, our system aims to present the expert with these two options from which s/he confirms the one fitting their intentions (say, the former interpretation is preferable):

Is Acid Soil a **typical** (common, observed,...) Chemical Soil Type?
Is Acid Soil a **sub category** (sub-type) of Chemical Soil Types?

We finally have information about subject of modification (Acid Soil, Acid Soil Types), the pattern instantiated (P:ClassInstatiation), and a natural language description of the interpretation (*Acid Soil is a typical Chemical Soil*

Type). These information bits are wrapped into another message and sent to the ontology maintenance space were experts are able to fine-tune this change.

More complex modelling choices may unravel, for example, an **RT** relation, relying on background knowledge coming from content patterns of the agricultural domain. In any case, the approach pivots around the understanding that an expert transmits of its own task through contextual answering to the system's questions. This enables us to achieve that both thesaurus and ontology are maintained without necessary mixing workers' competencies. Also we avoid to force any type of user to conform to a single way of carrying out maintenance, and to adopt a flexible way of setting correspondences among different knowledge models –alternatively to ad-hoc transformations.

4 Future work

Ethnographic study is ongoing at the moment; its objective is to draw equivalence and differences of reasons and requirements for changes in both thesaurus and ontology spaces. It will include a view on the workflow, people roles and performance scenarios formalized in to an knowledge base of patterns of maintenance jobs, to set up the first layer of our framework.

References

1. Philipp Cimiano and Johanna Völker. *Text2Onto*, pages 227–238. 2005.
2. Mathieu d'Aquin, Claudio Baldassarre, Laurian Gridinoc, Marta Sabou, Sofia Angeletou, and Enrico Motta. Watson: Supporting next generation semantic web applications, 2007.
3. Adam Funk, Valentin Tablan, Kalina Bontcheva, Hamish Cunningham, Brian Davis, and Siegfried Handschuh. *CLOnE: Controlled Language for Ontology Editing*, pages 142–155. 2008.
4. Dragan Gasevic, Dragan Djuric, Vladan Devedzic, and Violeta Damjanovi. Converting uml to owl ontologies. pages 488–489, New York, NY, USA, 2004. ACM.
5. K. Kotis and G. A. Vouros. Human-centered ontology engineering: The hcome methodology. *Knowledge and Information Systems*, 10:109–131, 2006.
6. A. Liang, B. Lauser, M. Sini, J. Keizer, and S. Katz. From agrovoc to the agricultural ontology service/concept server. an owl model for managing ontologies in the agricultural domain. *OWL workshop*, 2006.
7. Jesús Barrasa Rodriguez and Asunción Gómez-Pérez. Upgrading relational legacy data to the semantic web. pages 1069–1070, Edinburgh, Scotland, 2006. ACM.
8. M. Sini, B. Lauser, G. Salokhe, J. Keizer, and S. Katz. The agrovoc concept server: rationale, goals and usage. *Library Review*, 57:200–212, 2008.
9. V. Tablan, T. Polajnar, H. Cunningham, and K. Bontcheva. User-friendly ontology authoring using a controlled language. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC), Genoa, Italy, May*, 2006.
10. G. A. Vouros. Technological issues towards knowledge-powered organizations. *Journal of Knowledge Management*, 7:114–127, 2003.

Janus: Automatic Ontology Construction Tool

Ivan Bedini [1], Benjamin Nguyen [2], Georges Gardarin [2]

[1] Orange Labs,
42 Rue des Coutures,
14000 Caen, France
ivan.bedini@orange-ftgroup.com

[2] PRiSM Laboratory, University of Versailles,
45 Avenue des Etats-Unis,
78035 Versailles, France
{benjamin.nguyen, georges.gardarin}@prism.uvsq.fr

Abstract. The construction of an ontology for a large domain still remains an hard human task. The process is sometimes assisted by software tools that facilitate some parts of the ontology construction life-cycle. But often they do not propose a methodology that considers the automation of the entire process. In this paper we present a method for deriving an ontology automatically. Then, we introduce Janus, an implementation of this approach, for deriving automatically a skeleton of an ontology from XML schema files in a given domain. Janus also provides different useful views that can be used for a final revision by an expert.

Keywords. Ontology Creation, XML Mining, Application Integration.

1 Introduction

Over the past ten years, the Semantic Web wave has shown a new vision of ontology use for application integration systems. Researchers have produced several software tools for building ontologies (like Protégé [1] or OntoEdit [2]) and merging them two by two (like FCA Merge [3] or Prompt [4]) or producing alignments (like OLA [5], Mafra [6], S-Match [7], H-MATCH [8]).

As shown in [9] different reasons limit their adoption to the integration of internet and enterprise applications: (i) the lack of tools capable of extracting and acquiring information from a collection of XML files (the “de-facto” format for applications information exchange definition); (ii) the complexity of aligning and merging more than two knowledge sources at a time, which also is a task excessively consuming of computational time; (iii) existing ontology building methodologies, are human centric and are able to assist engineers just automating one or few parts of the entire process.

In this paper, we propose Janus, a tool for semi-automatic derivation of ontologies from XML schemas. It implements a new approach to ontology generation that

provides a solution to the limitations described above.

The aim of this short paper is to introduce Janus, how it works and to show some different views (produced by the tool) of the knowledge automatically acquired.

2 Approach and Methodology

In this section is provided a general view of the automation aspect of the ontology generation implemented by Janus.

Several methodologies for building ontologies exist, such as OTK [10], METHONTOLOGY [11] or DILIGENT [12], but they target ontology engineers and not machines. As far as we know, methodologies for automating the ontology generation process still have not been defined. As shown in [9], different tools provide the automation of some tasks, but only few of them define a complete automation procedure.

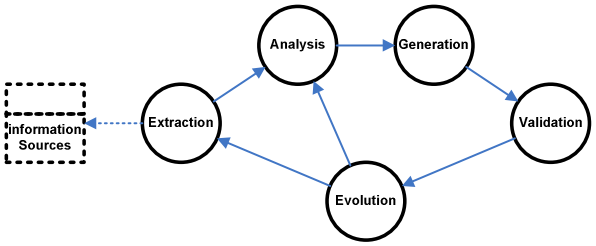


Figure 1 - Automatic ontology generation process

Following our experience we are now in process to define a methodology to automate the ontology construction. Briefly in this paper we present this methodology.

We define the automatic ontology generation life cycle as a cyclic process composed of five main essential steps to achieve the goal (see Figure 1). The main difference with a human centric methodology is that the "glue" between steps, as well as the integration of the different needed modules must be in a machine readable format. Also this approach is more dynamic and permits to acquire new inputs in order to maintain a reusable semantic memory and thus easily update the ontology constantly.

The underlying model that maintains the acquired information is based on RDF/OWL model. We have added some predefined relationships between concepts (like synonymy, have shared terms, have "a lot" of common properties, ...) in order to be able to define and maintain concepts similarities automatically retrieved. So doing, the generation step will look for concepts equivalence only between those concepts having at least one common link in order to be able to define a global ontology.

Furthermore, information about the confidence of the learned instances can be displayed with different views and used for a final revision by an expert. More details about the process are shown in [9].

3 XML as Input Source

XML schemas and ontologies in a given domain are somehow related. In general, schemas are built in a domain before ontologies. Consider for example the B2B domain: there exist hundreds of schemas to encode exchanged data but not many

ontologies. To benefit from preexisting schemas, we propose a method and a tool to derive an ontology or at least an enriched taxonomy (i.e., a concept hierarchy with concepts properties and main concepts similarity relationships) from a set of XML schemas. Janus currently implements a module for retrieving ontological information from this format, however its architecture is extensible and permits to add new modules retrieving information from other sources, like text documents or the Web.

4 Janus Architecture

Our tool implements an adaptation of several techniques originating from text mining and information retrieval / extraction fields, applied to XML files (that we call **XML Mining**). XML Mining is used to pre-process simple and compound statements defining XML tags, such as XSD elements and XSD complex types. It includes clustering methods based on a Galois Lattice and Formal Concept Analysis to quickly discovery similarities between names and structures extracted from the source corpora. Figure 2 shows the overall architecture of Janus.

The algorithm generating a high level representation of XML Schema information sources is composed of three main steps.

The first step is the **Extraction** task represented by the *Extract* arrow and *Acquisition* rectangle in Figure 2. It provides the knowledge needed to generate the ontology (concepts, properties and relationships). Implemented techniques for knowledge acquisition are a combination of different types, such as: NLP (Natural Language Process) for morphological and lexical analysis, association mining for calculating term frequencies and association rules, semantics for finding synonymy (implemented by the integration of an electronic dictionary like Wordnet), and clustering for grouping semantic and structural similar concepts.

The second step is **Analysis** represented by the correspondent block in Figure 2. This step focuses on the matching of retrieved information and/or alignment of two or more existing families of concepts issued from different input sources.

This step requires techniques already used in the first stage, as syntax and semantic measures, to establish the best similarities; it also requires an analysis of concept structures to determine hierarchical relationships and identify common properties. The output of this task provides enough information for building a semantic network of concepts that will be used in following step to look for similar concepts (it constitutes the basis of the semantic similarity memory of the system).

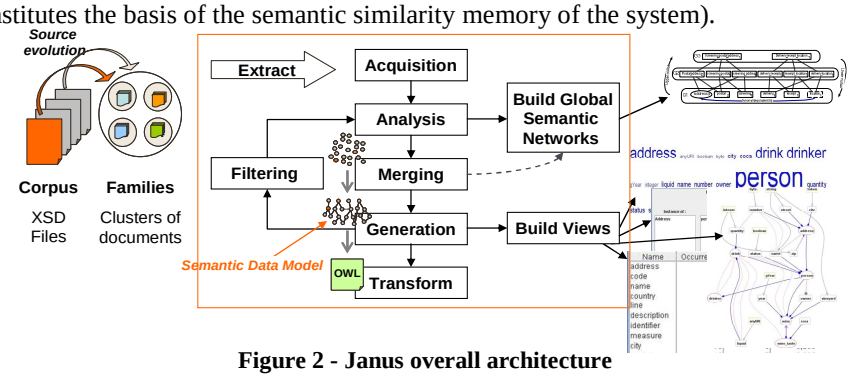


Figure 2 - Janus overall architecture

The last step is **Generation**, represented by the *Merging*, *Generation* and Filtering blocks in Figure 2. This step looks for concepts with evident affinities (e.g., concept fully included into another) based on specific rules, to merge or just link them. It generates a final semantic network that can be described in RDFS or OWL, built by the *Transform* module. The tool can derive from the network useful views provided to users by the *Build Views* module. Users can also step into the process to parameterize thresholds for refining results.

5 Functionalities and Views

The tool currently offers four visualization methods to view the acquired knowledge and a module able to generate a first ontology in OWL format.

The **word view** shows the list of terms composing the "ontology" as tag cloud format. The **detail view** shows all discovered relationships for a specific concept with other concept of the ontology. Between them we can find its properties shared in two main groups, "most common properties" and "other properties". This distinction permits to consider those properties characterizing the concept and the other that we can occasionally find for a concept. The **list view** gives detailed information about each concept like frequencies, family attendance and type (class, data-type or property). The **graph view** displays the semantic network of concepts (see Figure 3). The graph view can show the whole graph or only the part related to selected concepts with different layouts (hierarchical, tree, …).

Furthermore it is possible to select the kind of relationships to display. In fact acquired relationships are of different types: *propertyOf*, *synonym*, *shared terms* (compound tags with common terms like *address* for *tender_address* and *post_box_address*) and *relatedTo* (mainly merged concepts or other of type *owl:sameAs* and *owl:equivalentClass*). This feature permits to analyse in details some parts of the ontology; it is useful when the ontology is too large to be browsed with the global view.

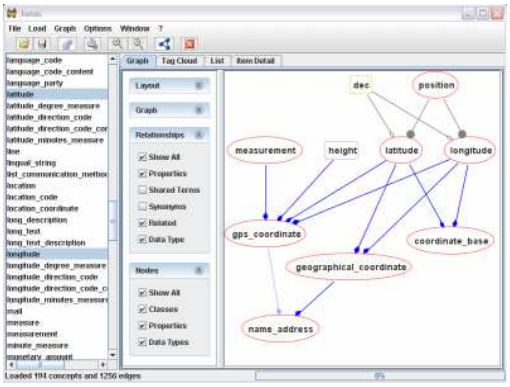


Figure 3 - Janus Detail Overview

Other views, one called "Concepts Social Network", and another to identify groups of common occurrences of properties, are under development.

Finally the generated ontology can be exported in OWL format. This is an important feature because permits to transform the Janus generated meta-model in a more generic format that can be used by other tool like Protégé [1].

The tool also offers the possibility to parameterize thresholds for alignment and merging operations.

6 Conclusion

The automatic construction of an ontology is a complex task that requires: i) a specific methodology capable to be executed autonomously by a machine; ii) an extensible semantic memory capable to easily discover concepts similarities and; iii) to be able to extract information from different sources.

We propose to demonstrate our preliminary results of Janus, a tool that we have developed to provide a first significant return of experience of a complete automation of the ontology construction. We will show our tool applied to the analysis of several B2B standards XML based, as input source. Differences between the presentation already done in [13, 14] and this one, are that our demo will focus on the automation methodology and, seeing that it is an ongoing work, we show results about new developed algorithms capable to better integrate the structure of input sources.

7 References

1. N. F. Noy, R. W. Fergerson, & M. A. Musen. The knowledge model of Protege-2000: Combining interoperability and flexibility. In: Proc. EKAW, France, 2000.
2. Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer and D. Wenke. OntoEdit: Collaborative Ontology Engineering for the Semantic Web. In: Proc. ISWC, Italy, 2002
3. Stumme, G., Maedche, A. FCA-MERGE: Bottom-Up Merging of Ontologies. In: Proc. IJCAI, Seattle, WA, 2001
4. N. F. Noy, M. A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In Proceedings of AAAI, 2000.
5. Jérôme Euzenat. An API for ontology alignment In: Proc. ISWC, Hiroshima (JP), 2004
6. Maedche, A., Motik, B., Silva, N., and Volz, R. MAFRA - Mapping Distributed Ontologies in the Semantic Web. In Proc. EKAW, 2002.
7. Fausto Giunchiglia, Mikalai Yatskevich, and Pavel Shvaiko. Semantic matching: Algorithms and implementation. Journal on Data Semantics, IX, 2007.
8. Silvana Castano, Alfio Ferrara, Stefano Montanelli. H-MATCH: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems. In: Proc. SWDB 2003
9. Ivan Bedini, Georges Gardarin and Benjamin Nguyen. Deriving Ontologies from XML Schema. In: Proc. EDA 2008. Toulouse, France, Juin 2008; RNTI, Vol. B-4, 3-17.
10. Sure, Y., Staab, S., Studer, R. On-To-Knowledge Methodology (OTKM). Handbook on Ontolo-gies, p117-132. 2004.
11. Corcho, O., Fernandez-Lopez, M., Gomez-Perez, A. Methodologies, tools, and languages for building ontologies. Where is their meeting point? Data Knowl. Eng. 46(1). 2003.
12. Vrandecic, D., Pinto., H. S., Sure, Y., Tempich, C. The DILIGENT Knowledge Processes. Journal of Knowledge Management 9 (5): p85-96. 2005.
13. Ivan Bedini, Benjamin Nguyen and Georges Gardarin. Janus: Automatic Ontology Builder from XSD files. In: Proc. WWW2008 - Developer track. Beijing, China, April, 2008
14. Janus demonstration video download: http://pagesperso-orange.fr/bivan/Janus

Guiding the Ontology Matching Process with Reasoning in a PDMS

François-Élie Calvier and Chantal Reynaud

LRI, Univ Paris-Sud & INRIA Saclay - Île-de-France
4, rue Jacques Monod - Bât. G
91893 Orsay Cedex FRANCE
{francois.calvier,chantal.reynaud}@lri.fr
http://www.lri.fr/iasi

Abstract. This article focuses on ontology matching in a decentralized setting. The work takes place in the MediaD project. Ontologies are the description of peers data belonging to the peer data management system SomeRDFS. We show how to take advantage of query answering in order to help discovering new mappings between ontologies, either mapping shortcuts corresponding to a composition of pre-existent mappings or mappings which can not be inferred from the network but yet relevant.

Key words: ontology matching, peer-to-peer, data management systems.

1 Introduction

Our work takes place in the setting of the MediaD project[1], which aims at creating a peer-to-peer data management system (PDMS) called SomeRDFS [1] allowing the deployment of very large applications that scale up to thousands of peers. We are interested in making the generation of mappings automatically supported by query answering. We propose to use query answering to generate mapping shortcuts and to identify relations, denoted target relations, which are starting points in the mapping discovering process. These relations allow identifying relevant mapping candidates limiting in that way the matching process to a restricted set of elements. Discovered mappings can be relevant or not according to the strategy involved in the PDMS. Indeed, a peer can decide to look for new mappings whatever they are (default strategy denoted S_1) or to look for particular mappings: either (strategy denoted S_2) new mappings involving peers already logically connected to it (there exists a mapping between their two ontologies) or (strategy denoted S_3) mappings involving peers not yet logically connected to it.

The paper is organized as follows. Section 2 shows how the query answering process can be used. Section 3 focuses on the identification of mapping candidates from target relations. We conclude and outline remaining research issues in Section 4.

[1] Research project funded by France Telecom R&D

2 Using Query Answering

2.1 Mappings Shortcuts Discovery

A mapping shortcut is a composition of mappings. Mapping shortcuts consolidate PDMSs by creating direct links between indirectly connected peers.We propose a two-step automatic selection process. We first identify potentially useful mappings shortcuts exploiting query answering. In this step, the goal is to retain only mappings which would be useful in the rewriting process with regard to the queries really posed by users to the peer $\mathcal{P}$. However, all these mappings will not be systematically added to the set of mappings of $\mathcal{P}$ because the usefulness of some of them may be low. Thus, we propose then a second selection step based on filtering criteria which can be different from one peer to another.

To achieve the first step we need to distinguish the rewriting and evaluation phases of query answering. Query answering will not be a unique and global process anymore but two connected processes which can be separated if needed. This separation allows to identify the relations that are interesting according to the user, i.e. the ones whithin the obtained rewritings he has chosen to evaluate.

The second selection step is based on the strategy of the peer and potentially exploits filtering criteria defined by the administrator of this peer. The usable criteria are specific to each peer but are limited. They concern either the kind of user who posed the query which originated the mapping (user-criterion) or the kind of relation belonging to $\mathcal{P}$ involved in the mapping (relation-criterion).

2.2 Identification of Target Relations Using Query Answering

In our approach we consider that a relation is a target relation if it is an obstacle for its peer in achieving the strategy it has chosen to implement. The definition of a target relation will then be based on a counting function. That function will differ according to the strategy of the peer and also according to the method used to count. The result of the counting function will be compared to a threshold that will be fixed by the administrator of the peer. When the value of the function is lower than the threshold the relation will be a target relation.

Definition (Target Relation) $\mathcal{P}_1{:}R_1$ is a target relation iff f($\mathcal{P}_1{:}R_1$) < t, f being a counting function and t a threshold.
In [2], we precise the definition of the function f for the relation R_1 of the peer $\mathcal{P}_1$ according to the strategy chosen by the peer and according to the method, C_1 or C_2, used to count. C_1 operates with regard to the knowledge of the peer, its ontology and its mappings. C_2 is based on rewritings obtained from queries.

If the strategy of $\mathcal{P}_1$ is S_1 the result of f($\mathcal{P}_1{:}R_1$) is the number of distant relations specializing R_1. If the strategy of $\mathcal{P}_1$ is S_2 the result of f($\mathcal{P}_1{:}R_1$) is the number of distant peers involved in the set of relations more specific than R_1. If the strategy of $\mathcal{P}_1$ is S_3, R_1 will be a target relation if there is at least one peer involved in a low number of specialization statements of R_1. Thus, f($\mathcal{P}_1{:}R_1$) provides the minimum number of relations of a given distant peer specializing R_1.

3 Obtaining a Set of Relevant Mapping Candidates

Target relations are used to identify a restricted set of mapping candidates according to two scenarios. In the first scenario, let us consider $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$ three peers with C_1, C_2 and C_3 three classes and the following mappings: $\mathcal{P}_1{:}C_1(X) \Rightarrow \mathcal{P}_2{:}C_2(X)$ and $\mathcal{P}_3{:}C_3(X) \Rightarrow \mathcal{P}_2{:}C_2(X)$, each known by the two involved peers. This scenario is represented Figure 1.

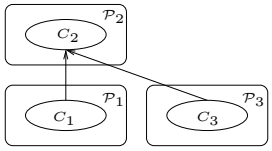


Fig. 1. Scenario 1

From the point of view of $\mathcal{P}_1$ $C_1(X)$ is a target relation . That target relation is interesting since $\mathcal{P}_1{:}C_1(X) \Rightarrow \mathcal{P}_2{:}C_2(X)$ is a mapping in $\mathcal{P}_1$, $Q_5(X) \equiv \mathcal{P}_2{:}C_2(X)$ could be a query posed to $\mathcal{P}_2$ by $\mathcal{P}_1$. The obtained rewritings would be $\mathcal{P}_1{:}C_1(X)$ and $\mathcal{P}_3{:}C_3(X)$ and looking for mappings between all the relations belonging to this set of rewritings is relevant.

In the second scenario let us consider $\mathcal{P}_1$ and $\mathcal{P}_2$ two peers, $\mathcal{P}_1{:}C_1$, $\mathcal{P}_2{:}C_2$ and $\mathcal{P}_2{:}C_3$ three classes. $\mathcal{P}_2{:}C_2(X) \Rightarrow \mathcal{P}_2{:}C_3(X)$ is a statement in $\mathcal{P}_2$. $\mathcal{P}_2{:}C_2(X) \Rightarrow \mathcal{P}_1{:}C_1(X)$ is a mapping in $\mathcal{P}_2$ and $\mathcal{P}_1$. This scenario is represented Figure 2.

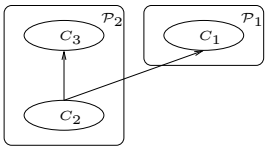


Fig. 2. Scenario 2

From the point of view of $\mathcal{P}_2$ $C_2(X)$ and $C_3(X)$ are target relations. This scenario is interesting since $\mathcal{P}_2{:}C_2(X) \Rightarrow \mathcal{P}_1{:}C_1(X)$ is a mapping in $\mathcal{P}_2$, it could be relevant to look for mappings between $C_1(X)$ and $C_3(X)$, two relations which subsume $C_2(X)$.

For each target relation we look for sets of mapping candidates, denoted MC. Our approach is based on the idea that it is relevant to look for connections between relations if they have common points. In our setting the common point that we are going to consider is a common relation, either more general or more specific. The construction of the set of mapping candidates can be achieved according to two processes, one for each scenario.

4 Conclusion

In this paper we have presented how SomeRDFS query answering can offer an automated support for discovering new mappings. In particular, we have shown that query answering in a decentralized setting can be used to select elements which are relevant to be matched when the number of elements to be matched is a priori huge and when no peer has a global view of the ontologies in the network. Our approach is based on query answering and filtering criteria. Future work will be devoted to the alignment process itself performed on each set of mapping candidates and relying on earlier work done in the group [3].

References

1. Adjiman, P., Chatalic, P., Goasdoué, F., Rousset, M.C., Simon, L.: Distributed reasoning in a peer-to-peer setting: Application to the semantic web. JAIR **25** (2006) 269–314
2. Calvier, F.E., Reynaud, C.: Guiding the ontology matching process with reasoning in a pdms. Technical Report 1495, CNRS-Université Paris Sud - LRI (June 2008)
3. Reynaud, C., Safar, B.: When usual structural alignment techniques don't apply. In: The ISWC'06 workshop on Ontology matching (OM-06). (2006)

Frame-based Ontology Learning for Information Extraction

Diego De Cao, Cristina Giannone, and Roberto Basili

University of Roma Tor Vergata, Italy, email: {decao,giannone,basili}@info.uniroma2.it

Abstract. In this paper, an ontology learning platform, called "*Frame-based Ontology Learning for Information Extaction*" (FOLIE), based on FrameNet, as a system of reusable knowledge patterns, the frames, and on lexical semantic primitives, i.e. word senses, is presented.

1 Introduction

It has been observed that ontology engineering can rely on general knowledge schemata highly reusable across domains and applications. *Conceptual* (or *Content*) *Ontology Design Pattern* (*CODeP*), as they have been called [7], impact in the ontology engineering phases as they can be customized through *specialization* or *composition* operators for modeling complex phenomena in a domain. Existing linguistic resources encode general (i.e. domain independent) information about a language. Building on the so called *frame* semantic model, the Berkeley FrameNet project [2] defines a frame-semantic lexicon for the core vocabulary of English. As defined in [6], a *frame* is a conceptual structure, modeling a prototypical situation and evoked in texts through the occurrence of its *lexical units* (LUs), i.e. predicates (such as nouns or verbs) that linguistically expresses the target situation. Lexical units of the same frame are predicates sharing the different semantic arguments, here called FRAME ELEMENTS, and constrained by a system of semantic types. FrameNet thus describes highly general and domain independent conceptual relations, tightly connected with the notion of *knowledge templates* that inspires CODePs. One of the advantages of frames is that they are firmly grounded on linguistic basis. As a methodological perspective, we see frames as a system of semantic relations for an ontological resource (as also explored in [11]). Similarly to CODePs, frames exhibit general semantic properties, but, from an ontology learning perspective, they can be directly employed to guide the process of knowledge acquisition from domain corpora. The approach followed in FOLIE is based on the assumption that semantic type constraints in a language can be usefully expressed in terms of a system of *word senses*, as encoded in lexical knowledge bases. The WordNet semantic dictionary for nouns will be thus used as a reference ontology for sense descriptions, as previously explored in [8]. This paper summarizes the general process for ontology learning from texts following the above assumptions and embodied in the FOLIE system. It derives domain specific frames as conceptual primitives useful for a target application. Frames here play the role of general CODeP and their specialization consists of three major steps: (1) *Assignment of lexical units to individual frames* (patterns) with the possibility of revising the general FrameNet definition by pruning existing LUs and discovering

novel LUs; (2) *Detection of the main roles for the domain patterns*, as specialized frame elements, whose relationships with the syntactic phenomena observed in the corpus is explicit; (3) *Specialization of semantic type constraints of individual roles* through a controlled generalization of the observed textual phenomena.

2 Ontology learning through unsupervised frame induction

The unsupervised acquisition of domain specific knowledge follows the process shown in Figure 1, where two main stages are foreseen. The first is *Discovery of Lexical Units*, responsible of generating an expressive geometric space from a corpus and supporting the assignment of novel lexical units to frames. The second stage is *Frame Acquisition*, in charge of inducing linguistic and ontological patterns by extracting corpus sentences relevant to a given frame, generalize them through Wordnet and locate them in Framenet. Here the syntactic dependencies exposed by predicates in the parsed texts are generalized and their selectional preferences mapped into WordNet concepts. These provide a number of semantic patterns that are then mapped into frames and frame elements. The compilation of the acquired information in OWL is then accomplished. Accordingly, FOLIE has been developed as a distributed system. Different tasks correspond to different, asynchronous, processes. The major components are developed in Java, whereas specific tools are written in C, as the SVD decomposition library[1]. Some individual steps foresee manual validation and a Web interface for each task has been developed. The overall process is hereafter summarized, while a more detailed description is reported in [4].

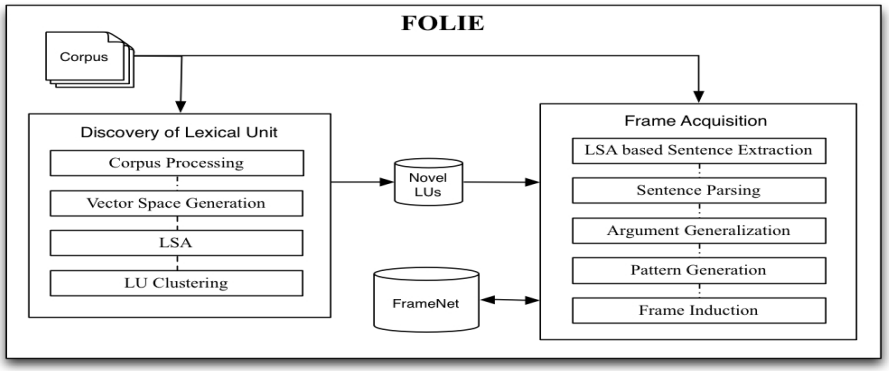


Fig. 1. The general inductive approach to the acquisition of frame-like knowledge

Discovery of Lexical Units In [12] a vector space model of *Frame Semantics* has been proposed. Its main assumption is that the notion of *frame*ness can be modelled and represented by a proper Semantic Space. The early phases of analysis in FOLIE are responsible for the construction of the Semantic Space. First, a basic vector space is extracted as a distributional model for individual words. The geometrical transformation

[1] URL: `http://tedlab.mit.edu/ dr/svdlibc/`

known as Latent Semantic Analysis (LSA)[10] is then applied. As words are represented as vectors of pseudo-concepts, lexical units are here used to locate frames in the resulting space induced by LSA. For each frame F, the known LUs for F are first clustered through an adaptive variant of the original *k-mean* algorithm, called *QT* clustering[9]. The number of clusters output by the algorithm is not fixed, and the initial k seeds may give rise to an arbitrary number of clusters, depending on the source data distribution. At the end of this phase for each frame F some clusters C_F are produced. A cluster C_F is thus a region of the space where the $frame$ness property manifests. Distance from a cluster (usually computed as the distance with respect to the cluster centroid) is a distributional criterion for deciding about the frame membership of words not yet known as lexical units for F. If the vector representation of a word (e.g. a verb) is close enough to the centroid of a cluster C_F derived for a frame F, it is selected as a novel candidate LU for F. In Fig. 2, the browsing interface for the validation of the LU classification is shown. The upper part shows the biplanar graph representing the LSA space triggered by the query verb *kill*: all the frames close to *kill* are shown. The lower part shows the cluster of the LUs for the closest frame, i.e. KILLING: they are the distributionally most similar words to the target word *kill*. In this case they are clear suggestions of the correctness of the frame KILLING.

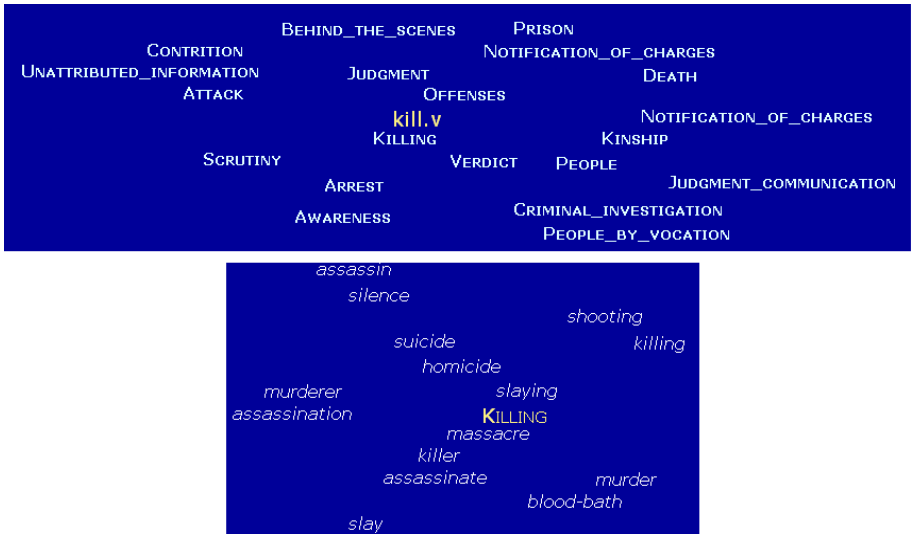


Fig. 2. The FOLIE toolbox: visualization of the LU space for the word *kill*.

Sentence Extraction and Filtering The sentence extraction task aims to detect portions of the corpus where a given frame is realized. Sentences are characterized by diverse local properties and deciding if a frame manifests in a text fragment is a more complex task than LU classification. A more fine-grained approach is required to exploit the locality properties in the LSA space and model the frame semantics of individual sentences. The duality property of LSA allows to represent terms and texts in the same k-dimensional space. Once a cluster C_F, representing a frame F, is found useful to classify a given target word tw in F, the same topological criteria can be used to decide about texts. Text vector representations in the LSA space can be obtained as linear

combinations of words (i.e. features). These are directly computed in the LSA space and those *close enough* to the cluster C_F are retained as candidate manifestations of F. Texts having similarity with the cluster centroid higher than a threshold can be retrieved from the LSA space, in analogy with the Latent Semantic Indexing process. Notice that this topological constraint can be coupled with a stricter rule that imposes the occurrence of some lexical units of F. The retrieval rule is thus the following:
(1) *Retrieve* all sentences *close* to centroid of the frame F AND
(2) *Filter out* those NOT including any lexical unit of F.

Argument Generalization The sentences related to frames extracted from the corpus provide the syntactic realisations of semantic arguments for the target predicate (frame). First, dependencies are extracted through parsing of the retrieved sentences. Individual LUs, nlu, appear in specific sets of sentences and syntactic analysis is run against these latter[2]. Parsed material defines the lexical fillers LF_r for the grammatical dependencies r activated by one LU. These are highly informative about the specific usage of a predicate F in the corpus and allow to acquire domain specific knowledge: the specialization of F consists here in the definition of a new (more specific) frame whose FEs are *all and only* those required by the corpus. FEs can be here induced from the sets LF_r through generalization. As a first step the best Wordnet types able to generalize the fillers have to be found. The utility function adopted here is the *conceptual density*, cd ([1, 3]). The greedy algorithm described in [3] allows to compute the minimal set of synsets (i.e. common hypernims for members of LF_r) that have the maximal conceptual density in Wordnet and cover all the fillers in LF_r. Every common hypernim α discovered by the above greedy algorithm suggests (at least) one sense for *some* words of LF_r, that is a possible semantic constraints for the dependency r.

Pattern Generalization After the generalization in WN is made available, the syntactic arguments for each nlu give rise to a set of pairs (r, α) where r is a syntactic relation and α is a WN synset. A full pattern is defined as a $n + 1$-ple:

$$((r_1, \alpha_1), nlu, (r_2, \alpha_2), ..., (r_n, \alpha_n))$$

where (r_i, α_i) are the i-th arguments as observed in full sentences. Complete patterns are derived by looking to sentences in corpus that contain all the arguments. After the previous phases, a large number of patterns as syntactico-semantic templates are made available for known LU or novel ones. They express the specific behavior of lexical units that are predicates (i.e. frames) in the corpus.

Frame Induction In order to compile domain specific frames for a lexical unit $nlu \in F$, its individual syntactic relations r and generalizations α in WN must be suitably mapped to frame elements. The reference knowledge for this task is given by both the different FEs, possible for a frame F, and their semantic type restrictions. In order to map (r_i, α_i) pairs to FEs, semantic disambiguation is applied again. The FE nominal head feh is extracted from its definition. For each pair (r, α) its semantic similarity with respect to all possible FEs is computed through conceptual density scores run over the set $LR_f \cup \{feh\}$. The FE that maximizes the cd score is selected as the correct interpretation of (r, α). The mapping of all dependencies foreseen by a pattern results in a specialized frame. The specialization process provides domain specific type constraints in terms of the WN-based synsets, α_i. In order to encode the resulting information, the

[2] The RASP parser ([5]) is here adopted.

OWL FrameNet resource (as proposed in [11]) is updated: new frames are introduced for each pattern and their FEs are constrained through the WN semantic types. In Figure 3 the proposed interpretations for the pattern $(\text{"}fire, attack\text{"}, kill, \text{"}family\text{"})$ are shown: by selecting the correct interpretation (MEANS *kill* VICTIM), the knowledge engineer triggers the OWL compilation of the corresponding specialized KILLING frame.

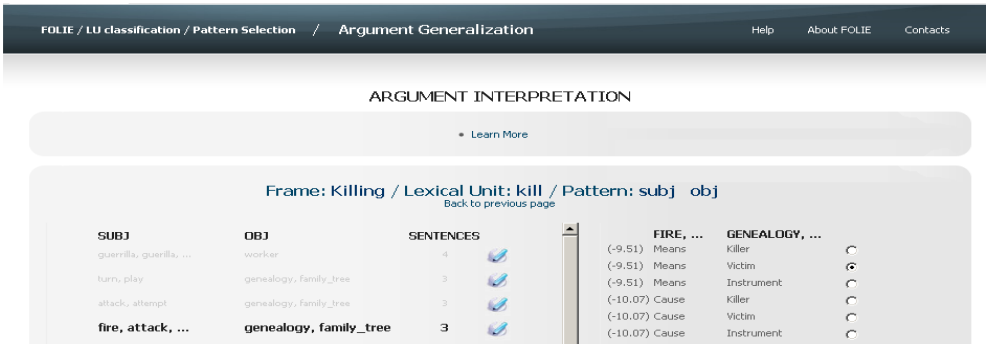


Fig. 3. The FOLIE toolbox: pattern acquisition for the verb *kill*.

References

1. E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *Proceedings of COLING-96*, Copenhagen, Denmark, 1996.
2. Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet project. In *Proceedings of COLING-ACL*, Montreal, Canada, 1998.
3. R. Basili, M. Cammisa, and F.M. Zanzotto. A semantic similarity measure for unsupervised semantic disambiguation. In *Proceedings of LREC-04*, Lisbon, Portugal, 2004.
4. R. Basili, C. Giannone, and D. De Cao. Learning domain-specific framenets from texts. In *Proceedings of ECAI-2008 Workshop on Ontology Learning and Population (OLP3)*, 2008.
5. E. Briscoe, J. Carroll, and R. Watson. The second release of the rasp system. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, Sydney, Australia., 2006.
6. Charles J. Fillmore. Frames and the semantics of understanding. *Quaderni di Semantica*, 4(2):222–254, 1985.
7. Aldo Gangemi. Ontology design patterns for semantic web content. In *Proceedings of the ISWC 2005, Galway, Ireland*, 2005.
8. Aldo Gangemi, Roberto Navigli, and Paola Velardi. The ontowordnet project: extension and axiomatization of conceptual relations in wordnet. In *" International Conference on Ontologies, Database and Applications of Semantics (ODBASE), Catania (Italy), 2003"*.
9. L.J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research*, (9):1106–1115, 1999.
10. Tom Landauer and Sue Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240, 1997.
11. Srini Narayanan, Charles J. Fillmore, Collin F. Baker, and Miriam R. L. Petruck. Framenet meets the semantic web: A daml+oil frame representation. In *"Proceedings of the The Eighteenth National Conference on Artificial Intelligence"*, 2002.
12. M. Pennacchiotti, D. De Cao, P. Marocco, and R. Basili. Towards a vector space model for framenet-like resources. In *Proceedings of LREC 2008*, 2008.

Ontology Engineering from Text: searching for non taxonomic relations in versatile corpora

Marie Chagnoux, Nathalie Hernandez and Nathalie Aussenac-Gilles

IRIT, University of Toulouse, 118 route de Narbonne,
31062 Toulouse Cedex 9, France

Abstract. In this paper, we propose a methodological approach based on pattern design and acquisition from texts in order to enrich lightweight ontologies with non-taxonomic relations. Since learning approaches require constrained domains and corpora with strong regularities, an alternative method is needed to locate sharp relations in versatile corpora. Rooted from our past experiments of Cameleon an ontology building tool, our approach relies on an existing ontology, an evolutive pattern base and a tagged corpus resulting from a morpho-syntactic analysis. The objective is twofold : (i) the morpho-syntactic patterns stored in the base are used to identify new relations between the concepts from the ontology (ii) new patterns identifying new kinds of relations are extracted from the context of co-occurring concept labels. These patterns enrich the pattern base and can be matched to look for new semantic relations.

1 Introduction

Relation extraction from texts can contribute to Ontology Engineering in extracting relations (or properties) between concept classes in Ontology Building. Since recent works rely on learning technics based on statistics combined with linguistics works or on experimented machine learning algorithms to support pattern-based relation extraction[1] [2], [3], we propose to consider automatisation more as an assisting process to the linguist than an independant task. Based on Hearst's ideas, we developed a first tool, Caméléon, that implements pattern matching in corpora to identify relations and concepts for ontology engineering. Using Caméleéon in different domains confirmed that manually tuning patterns and filtering matched sentences to identify conceptual relations is costly and time consuming [4]. This is particularly true for domain dependent relations. Our aim is to build on our experience in pattern-based relation extraction and in ontology building with Caméléon in order to preserve the sturdiness of extraction and improve the pattern acquisition process. We thus propose a methodological framework that better support the user during the pattern and relation identification tasks in the case where a hierarchy of concepts and their related terms are already available. In keeping with the options made in Caméléon, this framework promotes pattern reuse and adaptation. Its first novelty is to guide more efficiently the identification of related terms in the sentences matched with known patterns. The major change is to automatically suggest corpus-specific patterns: the system abstracts patterns from contexts of co-occurring terms that refer to concept labels. This paper will first present the bases of our consideration before introducing our framework -algorithm and implementation.

2 How to improve an ontology enrichment tool?

The method described in this paper is based on Caméléon. Caméléon provides assistance to reuse, adapt or design patterns for syntactically tagged texts. Then it supports pattern matching, human validation of the sentences found with the patterns that leads to defining terms and lexical relations, and finally it proposes an ontology editor where conceptual relations can be added.

Caméléon's ability to extract non-taxonomic relations from texts has been recently evaluated in [5] and [4]. These papers have shown both benefits and drawbacks of the approach. We decided to focus on the undeniable asset of the pattern-based approach which is its accuracy in discovering relations but we enrich the framework to assist the user. Because we assumed that a taxonomy of concepts and their related terms form the kernel of an ontology to be enriched with conceptual relations, we identified two possible means to improve this process: (i)using an existing pattern base to discover relations and enriching an existing ontology with these new relations (ii)learning patterns from the contexts of co-occurring terms or concepts from the ontology.

3 Algorithm and implementation

Figure 1 give an overviews of the system. The first part of the algorithm is dedicated to the processing - relations and pattern discovery - and the second one to the user validation[1]. For each pair of distinct ontology concepts (c_i, c_j), we look for all the

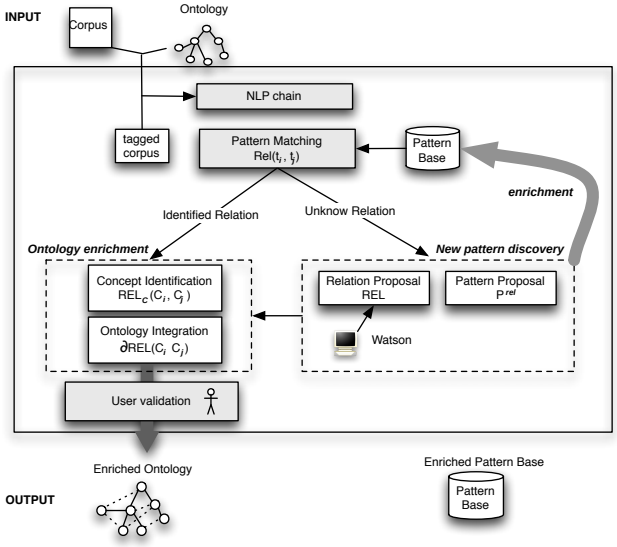


Fig. 1. Overview on our system

sentences that contain t_i, t_j where t_i, t_j belong respectively to the set of labels associated with concepts c_i, c_j. If one of the base patterns $P^{REL}i$ can be matched on the sentence s, we store the relation REL extracted by the pattern, the couple of concepts

[1] The validation phase is done in a second time in order to propose to the user all the couples extracted for a relation. We believe that this way the user's work is facilitated.

$REL^c(c_i, c_j)$ and s. If not, we search for a relation that could be defined in an existing ontology. If such a relation is found, we store the new relation noted REL_{new}, the couple of terms ${REL_{new}}^c(c_i, c_j)$ and s. For each new relation REL_{new} detected thanks to existing ontologies, we display to the user the relation and all the couples of concepts ${REL_{new}}^c(c_i, c_j)$. The user is asked to validate the relevance of the relation. If he validates the relation, the system proposes a set of patterns that can be generated according to the sentences where the couples have been identified. The user validates the relevance of the patterns. The patterns validated by the user are added to the base. To validate the relation, for each relation detected (REL and REL_{new}), the system displays to the user the labels of the relation and the couple of concepts related. The user then decides where to add the relation in the ontology. He can either decide to (i) add the relation between c_i and c_j; (ii) add the relation between an ancestor of c_i and an ancestor of c_j; (iii) add the relation between a concept linked to c_i in the ontology and a concept linked to c_i; (iv) reject the relation for the couple[2].

Contrary to learning approaches, the entire control of the user on the pattern construction process guarantees the semantic significance of the patterns and the relevance of the identified conceptual relations.

4 Conclusion

Pattern-based relation extraction from a corpus can be an efficient means to enrich an ontology. Provided patterns can be collected, accumulated, adapted and semi-automatically acquired, and provided related terms can be easily identified in matched sentences. To carry out this process, we propose a tool which extends the Caméléon relation extraction tool by integrating learning principles. This new tool, which is still being implemented embeds (i) term identification in phrases matching already-written patterns and (ii) a pattern-creation assistant based on automatic proposals but not on machine learning.

References

1. Alexander Schutz and Paul Buitelaar. Relext: A tool for relation extraction from text in ontology extension. In Y. Gil et al., editor, *ISWC 2005, LNCS 3729*, pages 593 — 606, 2005.
2. Philipp Cimiano. *Ontology Learning and Population from Text. Algorithms, evaluation and applications*. Springer, Berlin, 2007.
3. Christopher Brewster, Fabio Ciravegna, and Yorick Wilks. User-centred ontology learning for knowledge management. In Birger Andersson, Maria Bergholtz, and Paul Johannesson, editors, *NLDB*, volume 2553 of *Lecture Notes in Computer Science*, pages 203–207. Springer, 2002.
4. Nathalie Aussenac-Gilles and Marie-Paule Jacques. Designing and evaluating patterns for relation acquisition from texts with caméléon. *Terminology, special issue on Pattern-Based approaches to Semantic Relations*, 14(1):45 – 73, 2008.
5. Nathalie Aussenac-Gilles and Marie-Paule Jacques. Designing and evaluating patterns for ontology enrichment from texts. In Springer Verlag, editor, *EKAW 2006, 15th International Conference on Knowledge Engineering and Knowledge Management , Prague, oct. 2006*, pages 158 – 165, Prague, 2006.

[2] In order to facilitate his choice, for each couple the user can have access to the sentences where the couple have been found in the text and to the context of each concept in the ontology (labels of the concepts and related concepts).

Automatic Relation Triple Extraction by Dependency Parse Tree Traversing

DongHyun Choi and Key-Sun Choi

Computer Science Department
Semantic Web Research Center, KAIST
Daejeon, Korea
cdh4696@world.kaist.ac.kr,kschoi@cs.kaist.ac.kr

Abstract. To use the information on the web pages effectively, one of the methods is to annotate them to meet with ontology. This paper focuses on the technology of extracting relation triplets automatically by traversing dependency parse tree of a sentence in postorder manner, to build ontology from plain texts.

1 Introduction

Problem that prevents ontology from widespread using is that it is hard to build. To build ontology automatically, we need to acquire relation triplets from text automatically. Relation triplet acquisition can be divided into two procedures - relation triplet extraction and mapping triplets into one of predefined relations. In this paper, we propose a method to extract relation triplets from text, by traversing dependency parse tree using predefined rule sets.

Table 1. Relation extraction result of a sentence: "James visits a company which has held seminar in London."

Sentence: *James visits a company which has held seminar in London.*
Result:
Triple1: (*James*, *visit*, *company* AND (Process *holding* AND (Objective *Seminar*) AND (in *London*)))

2 Relation Extraction System

2.1 Overview

The overall architecture of this system is as follows: after parsing the given sentence using dependency grammar, seven preprocessing procedures are executed to give more information on decision tree for rule application to extract the relation triplets. After preprocessing, we traverse the resulting dependency tree in postorder to find the relation triplets by using predefined generic hand-written rule sets. In order to solve the long-distance problem, we need to transmit the information at the lower part of dependency tree to the upper part of dependency tree. To do that, we use RT(Reserved Term), RC(Reserved Clue) and RQ(Relation Queue). RT contains a single term which will be used as concept. RC contains a 'clue', which will be used to determine the kind of relation. RQ contains set of relation triplets which are extracted so far.

2.2 Preprocessing module

Term Marking In this phase, we mark terms in the dependency tree. Terms will be used as concept/instance in the resulting ontology.

Named Entity Recognition This phase assign the words to semantic information. Semantic information can be used to map the extracted triplets into the predefined set of relations.

Marking To-infinitive/Gerund Since To-infinitive/Gerund is a verb which is used as object of the other verb, we need to consider them differently from the other verbs.

Processing Coordinate Conjunction Coordinate conjunctions connecting verbs show two different cases in its dependency structure: (1)Two verbs share some contents(ex. subject), (2)Two verbs do not share any contents. For each case, we should change the parse tree so that we do not need to consider about coordinate conjunctions seperately.

Relative Anaphora Resolution Relative anaphora like 'which' or 'who' refers to another term in the sentence. Since we need to make relation with the term which is refered, not with 'which' itself, we need to resolve the relative anaphora.

Marking Action Action is a concept of ontology to be built, which represents action or status change of some object. We use Actions to gather two or more relation triplets which should be represented as a composite one. In table 2, result (1) does not mean that Samsung holds seminar in London - rather, it gives two partial informations which are wrong if they are not gathered. Thus, we use the concept of Action to get the triplet of result (2).

Merging Negation/Frequency with Verb Negation/Frequency information is merged with its attributed verb. Considering the sentence "James is not a student", we mark 'not' at the node 'is' so that the extracted relation triplet does not become (*James*, *ISA*, *student*).

Table 2. Relation Extraction from a sentence "Samsung has held seminar in London."

Sentence: *Samsung has held seminar in London.*
Without_Action – Result (1):
Triple1: (*Samsung*, *have hold*, *seminar*)
Triple2: (*Samsung*, *have hold in*, *London*)
With_Action – Result (2):
Triple3: (*Samsung*, *Process*, *Holding* AND (Objective *seminar*) AND (in *London*))

2.3 Dependency Tree Traversing Module

In this module, we extract relation triplets by post-order dependency tree traversing. Figure 1 shows the procedure of extracting relation triplets of sentence, "James visits a company which has held seminar in London", and the extracted triplets.

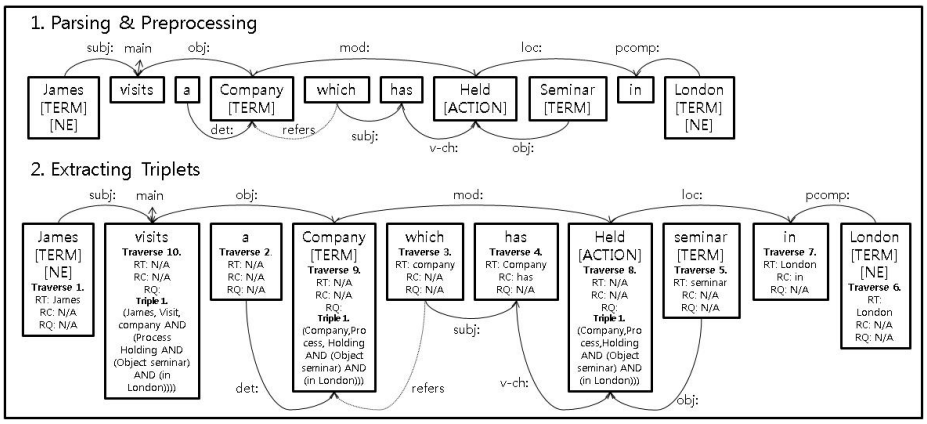


Fig. 1. Relation extraction example - step by step

3 Conclusion

This algorithm gives solution to long-distance problem, which cannot be solved using pattern matching method. Also, this algorithm extracts not only relation triplets but also the constraints of the arguments of triplets. This will surely enhance the quality of ontology built using the resultant triplets of this algorithm.

Acknowledgments. This work was supported in part by MKE & IITA through IT Leading R&D Support Project.

References

1. Hearst, M.A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: Proceedings of the 14th conference on Computational linguistics, pp.539-545. Association for Computational Linguistics, Nantes (1992)
2. Ravichandran, D., Hovy, E.: Learning Surface Text Patterns for a Question Answering System. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp.41-47. Association for Computational Linguistics, Pennsylvania (2001)

A Community Based Approach for Managing Ontology Alignments

Gianluca Correndo, Yannis Kalfoglou, Paul Smart, Harith Alani

University of Southampton,
Electronic and Computer Science Department
[gc3, y.kalfoglou, ps02v, h.alani]@ecs.soton.ac.uk,
WWW home page:http://ecs.soton.ac.uk
SO17 1BJ, United Kingdom

Abstract. The Semantic Web is rapidly becoming a defacto distributed repository for semantically represented data, thus leveraging on the added on value of the network effect. Various ontology mapping techniques and tools have been devised to facilitate the bridging and integration of distributed data repositories. Nevertheless, ontology mapping can benefit from human supervision to increase accuracy of results. The spread of Web 2.0 approaches demonstrate the possibility of using collaborative techniques for reaching consensus. While a number of prototypes for collaborative ontology construction are being developed, collaborative ontology mapping is not yet well investigated. In this paper, we describe a prototype that combines off-the-shelf ontology mapping tools with social software techniques to enable users to collaborate on mapping ontologies. Emphasis is put on the reuse of user generated mappings to improve the accuracy of automatically generated ones.

1 Introduction

The transformation of the Web from a mere collection of documents to a queryable Knowledge Base (KB) is one of the most prominent targets of Semantic Web (SW). To help reach this goal, knowledge repositories need to publish semantic representations of their data models to enable other machines to understand and query their content. To this end, much research and development has focused on building tools and capabilities for ontology and KB construction. However, support for distributed teams to remotely and continuously collaborate on building and updating ontologies and knowledge repositories is still underdeveloped. In this paper we describe an approach and present a prototype for facilitating ontology mapping by supporting social collaboration and reuse of mapping results for supporting data integration task. More specifically, our approach allows to: **align** local ontologies to shared ones; **exploit** social interaction and collaboration for improve alignment quality; **reuse** user ontology alignments for improving future automated alignments.

2 Collaboration for Knowledge Sharing

The need to make explicit, agree and publish data semantics is becoming increasingly central since more information systems are becoming largely decoupled and

separately managed. To this end, the vision of the SW is moving towards a scenario where the task of creating and maintaining ontologies, that formalise data semantics, is going to be handed to the community that actually uses them [1]. Such vision requires that latent models shared by the community must emerge and tools and methodologies must be provided for fulfilling this task.

The rise of Web 2.0 has transformed the classical community of passive Web users into a community of active contributors. Leveraging upon this new perspective of web communities, several proposals have lately emerged to exploit users' contributions for supporting various knowledge tasks[2].

Collaborative Protègè [3] was recently developed as an extension to Protègè to support users to edit ontologies collaboratively, by providing them with services for proposing and tracking changes, casting votes, and discussing issues, thus infusing classical ontology editing with a number of popular social interaction features.

Other Web 2.0 inspired approaches rely on *lighter* ontologies, where the emphasis is put on sharing knowledge rather than creating an ontology. Some of these approaches use social tagging as the main driver for enacting collaborative lightweight ontology building [4]. Similarly, other tools are focussing on editing and sharing instance data, like OntoWiki [5] and DBin [6].

Most of the tools listed above focus on supporting users to collaboratively construct ontologies or to collaboratively populate an ontology with instance data. Unlike these tools, our proposed system, OntoMediate, extends the collaborative notion to support the task of *ontology mapping*, where users can collaborate and interact to map their existing ontologies and reuse mapping structures. A similar approach is the Zhadanova and Shvaiko [7] method. Focus of that work was on building such profiles to personalise reuse of ontology mappings. In OntoMediate though, we are exploring the use of collaborative features (discussions, voting, change proposals) to facilitate the curation, reuse and discussion of mappings by the community, and hence paving the way to integrate distributed knowledge bases.

3 OntoMediate System Description

In the OntoMediate [1] project we are studying how social interactions, collaboration and user feedback can be used in a community, in order to ease the alignment of ontologies and to share mapping results[2]. The implemented prototype is a Web application developed with J2EE and AJAX technologies. The ontologies are expected to be written in OWL and Jena API[3] is used for parsing documents. The system has been designed to be extended via its APIs and is composed of three main subsystems: ontologies and datasets manager (section 3.1); ontology alignment environment (section 3.2); social interaction environment (section 3.3).

[1] http://www.ecs.soton.ac.uk/research/projects/ontomediate

[2] This work was partially funded by a grant awarded to General Dynamics UK Ltd. and the University of Southampton as part of the Data and Information Fusion Defence Technology Centre (DIF DTC) initiative.

[3] http://jena.sourceforge.net

3.1 Ontologies and Datasets Manager

This part of the system allows users to register (as well as unregister) the datasets they intend to share with the community and the ontologies that describe their data vocabulary. An ontology browser allows then to inspect usual information about managed ontologies (i.e. hierarchy of concepts, labels, annotations, descriptions, properties and constraints). The ontologies that are loaded onto the system need to be aligned with one or more shared ontologies in order to enable querying of the published data by the community.

3.2 Ontology Alignment Environment

The full automation of ontology alignment is not an easy task [8]. The factors that affect the computation and accuracy of ontology alignments are so delicate that we can not afford not to take into account user input. Our system provides an API for automated ontology alignment tools to be plugged in and also maintains data structures to store parameters needed by a particular tool to execute (e.g. threshold values or available tool options). The API allows an easy integration of new alignment tools by means of wrappers (already integrated tools are: CROSI mapping system [9], INRIA Align [10] and Falcon OA [11]). These tools support the alignment task by proposing to the user some initial candidate mappings. The results from different tools can be merged using a weighted mean of each contribution and the decision of which combination of tools to use can be parameterised together with the configuration used to invoke each tool.

Once the automated mapping has been executed, the results are displayed in a dedicated interface for review and for searching further alignments. The interface has two view modalities: *hierarchical* and *detailed*. In the *hierarchical* view the two taxonomies are faced and mapped concepts are highlighted. The user can browse both taxonomies and create new mappings by dragging a source concept and dropping it into a destination concept. In the *detailed* view the description of two focused concepts are faced and the user can inspect the descriptions and map the properties using the same drag & drop facility used for mapping the concepts. The users can alternatively **accept** or **reject** some automatically proposed mappings. This choice will be recorded by the system and will be used to filter future mappings towards this target concept, thus increase future ontology alignment *precision*.

3.3 Social Interaction Environment

This functionality allows users of a community that deal with similar data - and therefore have a mutual interest to maintain good quality alignments - to socially interact with each other. The aim of the social interaction is to exploit community feedback in order to enhance the overall quality of the ontology alignment and achieve agreement on semantics of concepts by means of community acceptance. This subsystem proposes to the user three views: **Ontology View**; **User View** and **Forum View**. The **Ontology View** (see Figure 1 top-left corner) displays an enhanced taxonomy browser for the selected shared ontology. The enhancements concern the user activities affecting the shared concepts, visualising additional information (e.g. concepts that have some incoming mappings are

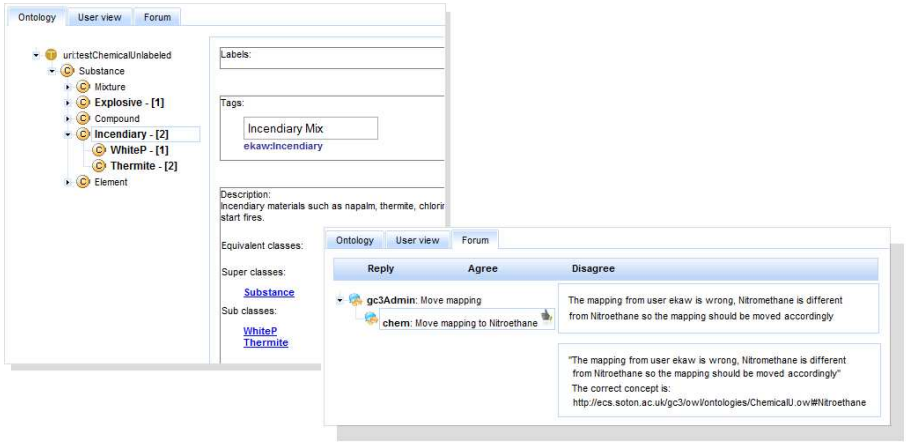


Fig. 1. Discussion environment - Ontology View

highlighted and the number of mappings is reported in brackets). Moreover, the interface allows to inspect the set of labels used for equivalent concepts (i.e. the ones provided with the alignments) in local ontologies (see the *Tags* text field in Figure 1). The user or administrator can edit such labels and add them to the shared concept to enrich the concept description with users' contributions. The new mapping, and the edited/added labels, will be logged in a database to be reused later to improve the *recall* of future ontology alignment tasks. When the user selects a concept that has some user mappings associated with it, he/she can switch to the **User View** that displays information about the local mappings for the focused concept. The user can then inspect a summarised description (i.e. subconcepts, superconcepts, properties etc.) of the local concepts and decide if they are relevant to the target concept or initiate a discussion thread in the forum (see Figure 1 bottom-right corner) in order to change them. Interacting within the forum users can debate the proposal, **reply** with a new one or simply **agree** or **disagree** with it. Relevant events are notified interested users (e.g. all the users that provided a mapping towards this target concept and all the others who explicitly asked to be informed).

3.4 Ontology Mapping Reuse

In OntoMediate system, one of the aims is to reuse user inputs in order to increase the quality of data integration and ease the ontological alignment task. Our approach for fulfilling this goals is twofold and involves user alignment results as an important input for increasing system performances and sharing achieved alignments.

The adoption of lexical information from local concept and properties for enriching target entities' description is just an example of how local contributions can help in building up a community tailored ontology. Such approach have

shown, based on preliminary tests, to increase the performances of automated tools for successive alignment tasks.

Moreover, the system provides an additional functionality that allows to seamlessly share the agreed ontology alignments by means of POAF (Portable Ontology Aligned Fragments) [12]. POAF uses existing alignments and OWL taxonomic reasoning to identify fragments that can be reused as minimal information bundles for building an integrated data network.

4 Summary and Future Work

This paper presented a prototype for supporting ontology mapping with community interactions, where users can collaborate on aligning their ontologies. Some initial experiment on reuse of lexical information from mappings showed an increase in both precision and recall in ontology mapping when reusing past mapping results. Next, we plan to run much larger experiments to further test the validity of the social approach, and the usability of the services and features provided. We will also implement services to allow users to submit and manage more complex mapping relationships.

References

1. Shadbolt, N., Berners-Lee, T., Hall, W.: The semantic web revisited. Intelligent Systems, IEEE **21**(3) (2006) 96–101
2. Correndo, G., Alani, H.: Survey of tools for collaborative knowledge construction and sharing. In: Workshop on Collective Intelligence on Semantic Web (CISW 2007). (November 2007)
3. Tudorache, T., Noy, N.: Collaborative Protégé. In: Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007) at WWW 2007, Banff, Canada (2007)
4. Zacharias, V., Braun, S.: SOBOLEO – social bookmarking and lightweight engineering of ontologies. In: Proc. WWW 2007 Workshop on Social and Collaborative Construction of Structured Knowledge, Banff, Canada (May 2007)
5. Auer, S., Dietzold, S., Lehmann, J., Riechert, T.: OntoWiki: A tool for social, semantic collaboration. In: Workshop on Social and Collaborative Construction of Structured Knowledge (CKC) at WWW 2007, Banff, Canada (2007)
6. Tummarello, G., Morbidoni, C., Nucci, M.: Enabling semantic web communities with DBin: An overview. In: Proc. 5^{th} Int. Semantic Web Conf., ISWC 2006, Athens, GA, USA. (2006)
7. Zhdanova, A.V., Shvaiko, P.: Community-driven ontology matching. In: ESWC. (2006) 34–49
8. Kalfoglou, Y., Schorlemmer, M., Uschold, M., Sheth, A., Staab, S.: Semantic interoperability and integration. Seminar 04391 - executive summary, Schloss Dagstuhl - International Conference and Research Centre (September 2004)
9. Kalfoglou, Y., Hu, B., Reynolds, D., Shadbolt, N.: Capturing, representing and operationalising semantic integration (CROSI) project - final report (October 2005)
10. Euzenat, J.: An api for ontology alignment. In: Proc. 3^{rd} Int. Semantic Web Conf. (ISWC), Hiroshima ,Japan (2004)
11. Jian, N., Hu, W., Cheng, G., Qu, Y.: Falcon-AO: Aligning ontologies with falcon. In: Workshop on Integrating Ontologies (K-CAP 2005). (2005) 85–91
12. Kalfoglou, Y., Smart, P., Braines, D., Shadbolt, N.: POAF: Portable ontology aligned fragments. In: Proc. of the ESWC'08 Int. Workshop on Ontologies: Reasoning and Modularity (WORM'08), Tenerife, Spain. (2008)

Evaluating Ontology Modules Using Entropy

Paul Doran[1], Valentina Tamma[1], Luigi Iannone[2], Ignazio Palmisano[1]
{pdoran, v.tamma, i.palmisano}@liverpool.ac.uk,
iannone@cs.man.ac.uk

[1] Department of Computer Science, University of Liverpool, UK
[2] Department of Computer Science, University of Manchester, UK

Abstract. Ontology modularization has been the focus of much research recently; many techniques to carry out ontology modularization have been developed. This creates a problem in evaluating the results of the techniques. Ontology modularization techniques cannot solely be evaluated by examining their logical properties. Certain applications of ontology modularization, such as ontology reuse, require a new objective way to evaluate the results. This paper motivates the use of an entropy inspired measure to evaluate ontology modules.

1 Introduction

Ontology modularization has received much attention recently. This focus has largely been on the creation of techniques to carry out ontology modularization, and specifying the conditions for including or excluding elements of an ontology module. There is no objective way to assess the quality of an ontology module obtained by these techniques, thus making a comparative analysis very difficult.

Size, w.r.t. the number of concepts, has been used as a factor to evaluate the results of the ontology modularization tools. This is not an appropriate measure because it does not tell us anything about the contents of the ontology module. If size were to be the deciding factor on our choice of ontology modularization technique then the optimum sized ontology module would be of size 0. Doran et al[1] apply a precision and recall measure to evaluate their results. Whilst, these may be more suitable than size they only consider the hierarchy of the ontology.

These measures alone do not help an Ontology Engineer to assess the quality of an ontology module; or to carry out an objective evaluation across the techniques. An Ontology Engineer evaluates an ontology via subjective criteria, but all computed measures are objective and do not reconcile easily with the subjective criteria [2].

The solution proposed is to use an entropy inspired measure. Entropy in terms of ontologies can be equated to a notion of information content; and in turn information content can be linked to a notion of usability and reusability.

2 Entropy Based Measure

Entropy orginated in Physics and is central to the second law of thermodynamics. Shannon took this notion and applied it to information theory[3]. Shannon defines entropy as a measure of the average information content the recipient is

missing when he does not know the value of a random variable. The formula for entropy is:

$$H(X) = -\sum_i p(i) \log p(i)$$

Calmet & Daemi adapt the notion of entropy for measuring the entropy of an ontology [4]. The probability mass function $p(i)$ used by Calmet & Daemi is shown below.

$$p(i) = \frac{deg(i)}{\sum_{v \in V} deg(v)}$$

Improved Entropy Measure This entropy formula has some limitations because all edges are treated as equal and their direction is not taken into account; therefore the semantics of the ontology is not fully reflected by the entropy measure. This suggests that to overcome these limitations of the existing entropy measure direction has to be considered and different edges need to be treated differently. Thus, we can obtain a more fine-grained entropy measure by considering direction and splitting the entropy measure in two:

Language level entropy - This level is concerned with the edges that represent language level constructs.

Domain level entropy - This level is concerned with the domain specific edges.

Splitting Entropy The model is an edge-labelled directed multigraph G, given two alphabets Σ_L and Σ_D, that is a pair $G = (V, E)$ where:

- V is a finite set of vertices, $E = L \cup D$.
- $L \subseteq V \times \Sigma_L \times V$ is a ternary relation describing language level edges.
- $D \subseteq V \times \Sigma_D \times V$ is a ternary relation describing domain level edges.

$\Sigma_L = \{l_1, ..., l_n\}$ and $\Sigma_D = \{d_1, ..., d_n\}$ are sets of labels which will label the edges of L and D respectively. To label the respective edges of L and D we use the following functions: $label_l(L) = L \rightarrow \Sigma_L,\ label_d(D) = D \rightarrow \Sigma_D$

Language Level Entropy - $H_L(X)$ The language level entropy ($H_L(X)$) calculates the entropy associated with the language level edges. Let $G_L = (V, L)$ whit $G_L \subseteq G$, assuming all language level edges are equal, $p(i)$ is:

$$p(i) = \frac{degOut(i)}{|L|}$$

Where $degOut() = V \rightarrow \mathbb{R}$ for each v that exists in V such that $degOut(v) = |L_v|$ where $L_v = \{(v \times l \times x) | v \in V\}$

Domain Level Entropy - $H_D(X)$ The domain level entropy ($H_D(X)$) calculates the entropy associated with the domain level edges. We consider $G_D = (V, D)$ where $G_D \subseteq G$. We assume that elements of Σ_D that appear more frequently in D split their information content evenly, thus the weight associated

with the edge should be lower. Thus, for every $d \in D$ we have the following weighting function: $w() = \Sigma_D \rightarrow \mathbb{R}$ This assigns a real number to every element of the alphabet Σ_D such that: $w(d) = \frac{1}{|D_d|}$ where $D_d = \{(x \times \sigma_D \times x) | label_d(d) = \sigma_D\}$ This normalises the weights of the edges between 0 and 1. Thus, the $p(i)$ to compute $H_D(X)$ is:

$$p(i) = \frac{weightsFromNode(i)}{\sum_{v \in V} weightsFromNode(v)}$$

Where $weightsFromNode() = V \rightarrow \mathbb{R}$ for each v that exists in V such that $weightsFromNode(v) = \sum_{f \in F} w(f)$ where F is the set of edges from D involving v. Thus, we sum the weights of the edges outgoing from v and divide this by the sum of the weights of the outgoing edges for all elements of V.

Recombining The Entropy Measure The entropy measure has been split into two distinct measures; these are recombined to compute $H(X)$:

$$H(X) = H_L(X) + H_D(X)$$

Depending on the semantics encoded in the graph it may be necessary to consider $\top$ and $\bot$. Assuming that $\top$ and $\bot$ are elements of V then they will be taken into account in the above formula. However, you may just wish to consider the entropy amongst the user declared elements of V, as $\top$ and $\bot$ are usually required elements of the language (e.g., OWL). Therefore, the entropy measure would be:

$$H(X) = (H_L(X) + H_D(X)) - (H(\top) + H(\bot))$$

3 Future Work

There is a need to carry out an in depth study which compares existing measures used within ontology evaluation to both size and entropy inspired measures, to identify the measures which are crucial to Ontology Engineers when they are evaluating ontology modules.

Acknowledgements This work was supported by EPSRC and by the EPSRC funded project 'Evaluating ontologies for open agent environments'.

References

1. Doran, P., Tamma, V.A.M., Iannone, L.: Ontology module extraction for ontology reuse: an ontology engineering perspective. In: Proc. of 16th Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal. (2007) 61–70
2. Yu, J., Thom, J.A., Tam, A.M.: Ontology evaluation using wikipedia categories for browsing. In: Proc. of 16th Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal. (2007) 223–232
3. C.E.Shannon: A mathematical theory of communication. Technical Report 27:379-423, 623-656, Bell System Technical Report (1948)
4. Calmet, J., Daemi, A.: From entropy to ontology. In: AT2AI-4 - Fourth International Symposium "From Agent Theory to Agent Implementation" at the 17th European Meeting on Cybernetics and Systems Research (EMCSR), Vienna, April 2004. (2004)

A Framework for Schema-based Thesaurus Semantic Interoperability*

E. Francesconi, S. Faro, E. Marinai, M.A. Biasiotti, and F. Bargellini

Institute of Legal Information Theory and Techniques
Italian National Research Council (ITTIG-CNR)
{francesconi,faro,marinai,biasiotti,bargellini}@ittig.cnr.it
http://www.ittig.cnr.it

Abstract This work proposes a formal characterization of the schema-based thesaurus mapping problem as well as a specific approach within such framework on a case study aimed at mapping five thesauri of interest for European Union institutions.

1 Introduction

In the last few years accessing heterogeneous data sources in a distributed environment has become a problem of increasing interest. In this scenario the availability of thesauri or ontologies is an essential pre-condition to guarantee quality in document indexing and retrieval, therefore interoperability among thesauri is important to guarantee cross-collections retrieval quality [1]. This work proposes a methodological framework for semantic mapping between thesauri as well as a specific approach within such framework on a case study aimed at mapping five thesauri (EUROVOC, ECLAS, GEMET, UNESCO Thesaurus and ETT) of interest for the European Union institutions having only schema information available.

2 A formal characterization of the schema-based thesaurus mapping problem

Thesaurus mapping for the case-study is a problem of terms alignment where only schema information is available (*Schema-based mapping*) [2] [3]. It can be considered a problem where to measure the conceptual/semantic similarity between a term (simple or complex) in the source thesaurus and candidate terms in a target thesaurus. We propose to characterize the schema-based Thesaurus Mapping ($\mathcal{TM}$) problem as a problem of Information Retrieval ($\mathcal{IR}$). As in $\mathcal{IR}$ the aim is to find documents, in a document collection, better matching the semantics of a query, similarly in $\mathcal{TM}$ the aim is to find terms, in a term collection (target thesaurus), better matching the semantics of a term in a source thesaurus.

* This work has been developed within the tender n. 10118 "EUROVOC Studies" of the Office for Official Publications of the European Communities (OPOCE).

The $\mathcal{TM}$ problem can be formalized as $\mathcal{TM} = [D, Q, F, R(q_i, d_j)]$ where:

1. D is the set of the possible *logical views* of a term in a target thesaurus (documents representation in $\mathcal{IR}$);
2. Q is the set of the possible *logical views* of a term in a source thesaurus (queries representation in $\mathcal{IR}$);
3. F is the framework of term representations;
4. $R(q_i, d_j)$ is a ranking function, which associates a real number with (q_i, d_j) where $q_i \in Q$, $d_j \in D$, giving an order of relevance to the terms in a target thesaurus with respect to a term of the source thesaurus.

This framework can be implemented using RDF/OWL standards to represent concepts and relationships; in particular the standards SKOS (Simple Knowledge Organisation System) can be used.

3 A Thesaurus Mapping Case Study

According to the project specifications, a mapping between EUROVOC and the other thesauri of interest is expected. The basic mapping methodologies are applied to *descriptors* within corresponding microthesauri in their *English version* as a pivot language. The steps of the system workflow is here below described.

a) **SKOS Core transformation and terms pre-processing**
Thesauri XML proprietary formats are transformed into an RDF SKOS Core representation using XSLT techniques. To reduce the computational complexity, terms are normalized so that digits and non-alphabetic characters are represented by a special character; then *stemming* and *stopwords elimination* are performed.

b) **Term logical views in source (Q) and target (D) thesauri**
Term semantics is conveyed by its morphological characteristics, by the context in which it is used as well as by the relations with other terms. Therefore we propose to represent the semantics of a thesaurus term by (i) its *Lexical Manifestation*: a string of characters normalized according to pre-processing steps (the framework F is represented by strings and standard operations on strings); (ii) by its *Lexical Context*: a term vector d of binary entries (statistics on terms to obtain weighted entries are not possible since document collections are not available) composed by the term itself, relevant terms in its definition and linked terms of a T-dimension vocabulary (F is T-dimensional vectorial space and linear algebra operations on vectors); (iii) by its *Lexical Network*: a direct graph where nodes are terms and the labeled edges are semantically characterized relations between terms (F is the algebra operations on graphs).

c) **The proposed Ranking Functions (R)**
A ranking function R is able to provide a similarity measure between terms. R for *Lexical Manifestations*: Levenshtein distance/similarity applied on pre-processed strings normalized with respect to the longest string (therefore this measure varies in the interval [0,1]). R for *Lexical Contexts*: Correlation between

such vectors, quantified as the cosine of the angle between these two vectors. R for *Lexical Networks*: Graph Edit Distance, namely the minimum number of nodes and edges deletions/insertions/substitutions to transform a graph g_1 into a graph g_2. Because of computational complexity we have considered three variants of the Graph Edit Distance: the *Conceptual similarity* expressing how many concepts two graphs have in common; the *Relational similarity* indicating how similar the relations between the same concepts in both graphs are; the *Graph similarity* [4] expressing the number of nodes and edges shared by two graphs over the number of nodes and edges in a reference graph.

d) **Ranking among candidate terms and mapping implementation** Candidate terms of the target thesaurus are ranked according to the similarity measure values ($sim \in [0, 1]$) and a semantics to mapping relations is assigned using proper heuristic threshold values (T_1, $T_2 \in [0, 1]$) to decide exactMatch ($sim < T_1$), partial match (broadMatch or narrowMatch) ($T_1 < sim < T_2$) or No Match ($T_2 < sim$).

4 Interoperability assessment through a "gold standard"

Interoperability between thesauri has been assessed on a "gold standard" data set, namely the ideal set of expected correct term mappings. The "gold standard" produced by experts includes 624 relations (346 are exactMatch). System mapping performances have been assessed with respect to the "gold standard" using the system *Recall* since the automatic mapping is addressed to identify matching concepts within the system predictions, to be validated by humans. Preliminary experiments showed satisfactory performances to identify relations expressing generic association between terms (untypedMatch); good performances have been obtained as regards exactMatch relations, while the distinction between narrowMatch and broadMatch revealed a high degree of uncertainty. The proposed term logical views and related ranking functions outperformed a simple string matching between terms. In particular for EUROVOC vs. {ETT, ECLAS, GEMET} the Lexical Manifestation logical view and the Levenshtein Similarity ranking function gave the best results (untypedMatch Recall = 66.2%, exactMatch Recall = 82.3%), while for EUROVOC vs. UNESCO Thesaurus the Lexical Network logical view and the Conceptual Similarity ranking function gave the best results (untypedMatch Recall = 73.7%, exactMatch Recall = 80.8%).

References

1. M. Doerr, "Semantic problems of thesaurus mapping," *Journal of Digital Information*, vol. 1, no. 8, 2001.
2. E. Rahm and P. Bernstein, "A survey of approaches to automatic schema matching," *Int. Journal on Very Large Data Bases*, vol. 10, no. 4, pp. 334–350, 2001.
3. J. Euzenat and P. Shvaiko, *Ontology Matching*. Springer, 2007.
4. W.-T. Cai, S.-R. Wang, and Q.-S. Jiang, "Address extraction: a graph matching and ontology-based approach to conceptual information retrieval," in *Proc. of the Third Int. Conference on Machine Learning and Cybernetics*, pp. 1571–1576, 2004.

Comparing background-knowledge types for ranking automatically generated keywords

Luit Gazendam[1], Véronique Malaisé[2], Hennie Brugman[3], and Guus Schreiber[2]

[1] Telematica Instituut, Enschede, The Netherlands
[2] Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands
[3] MPI for Psycholinguistics, Nijmegen, The Netherlands

1 Introduction

The automatic generation of thesaurus keywords can be a precious help to cataloguers working in large, daily growing archives. Given a text in which we spotted all lexical variants of thesaurus keywords, we face the problem of ranking the automatically generated keywords in order to suggest only a small list of most relevant keywords. Of course we could use the TF.IDF ranking, a classic, count-based ranking. We experiment in this paper wether we can improve upon the classic, count-based ranking of TF.IDF by using background knowledge which is represented in the relations between keywords. So we implemented two ranking algorithms that take into account the relations the keyword has to other found keywords. Next to the two ranking algorithms, we also tested the value of two types of background knowledge. We conducted the research within the archives of the Dutch institute for Sound and Vision.

2 Experiment

2.1 Material: background knowledge sources and test corpus

At Sound and Vision, we identified two background knowledge sources which model relationships between keywords: the archives thesaurus (named GTAA) and the archives catalogue, from which we extracted a co-occurrences network (Co-oc). The GTAA provides a general model of the world. It contains 3800 keywords which can be used to describe the subject of TV programs. These keywords are organized in hierarchical relationships (broader term/narrower term relation), associative relationships (related term, between keywords that belong to the same domain such as *planes* and *kerosene*)) and linguistic relationships (use/use for, between keywords representing the same notion). Each keyword averagely has 1 broader, 1 narrower and 3.5 related terms. The co-occurrence network shows what keywords are used together in practice. This network connects many more keywords (19 co-occurrence relations per keyword on average), but the semantics of this co-occurrence relationship is unspecified; a manual analysis showed us that it was often a loose associative relationship.

Our textual corpus contains 362 documents, referring to 258 catalogue descriptions. These catalogue descriptions contain keywords which were assigned

manually by cataloguers from Sound and Vision. These keywords are the ground truth against which we evaluate the TF.IDF baseline and the four possible combinations of ranking algorithms and background knowledge sources.

2.2 Ranking algorithms

We used GATE[1] and its plug-in Apolda[2] to extract possible references to the keywords for a TV-program. The TF.IDF of all these references to keywords is computed as a baseline ranking. For one TV-program, some of the found keywords have relations to other found keywords. Together the keywords and the relations form a graph. To increase the connectedness of our graph we also included indirect relations (in which an intermediate keyword connects two found keywords). The GTAA-relations and the co-occurrence network, when used as input for generating these graphs result in two different outputs.

We implemented two algorithms which transform this graph into a ranked lists: our own method called CARROT and the well known algorithm named Pagerank [3]. CARROT uses only the local connectedness of keywords. It creates four groups each having the same local connectedness and sorts each group on the TF.IDF values. Pagerank ranks keywords based on the entire graph structure. It computes for each keyword a Pagerank, which correspond to the eigenvector of the transition matrix of the graph (*i.e.* the structure of the relations). The Pagerank score expresses the importance of each keyword in the graph. We benchmark the four possible combinations (CARROT+GTAA, CARROT + co-oc, Pagerank+GTAA, Pagerank+co-oc) against the TF.IDF baseline.

2.3 Results

We evaluated the five settings against the manually assigned keywords. This evaluation was performed against a requirement of *conceptual consistency*[4]. This allows keywords which present a semantic similarity with the manually assigned keywords to be counted as correct too. This requirement better suits the cataloguer needs. In figure 1 the precision recall numbers are displayed.

The best method is CARROT + GTAA. The area which is the most valuable for cataloguers is the top of the suggestion list. In this area the CARROT algorithm performs the best. Only at rank 11 the Pagerank algorithm with the GTAA graph as input overtakes CARROT. At that time the recall is 70%.

The baseline (TF.IDF ranking) and CARROT with co-occurrences are both consistently worse then the CARROT with the GTAA setting. All GTAA based settings are better than the co-occurrence based settings. This means that even the loose associative relationship of the co-occurrence network contains enough information to improve the results. The contribution of the background knowledge to the performance is less big however than the contribution of the ranking algorithm (CARROT or Pagerank). Pagerank is much worse than CARROT. But the average precision of the Pagerank methods drops less with an increase in recall. This allows both Pagerank methods to overtake the other methods from suggestion 10 onwards.

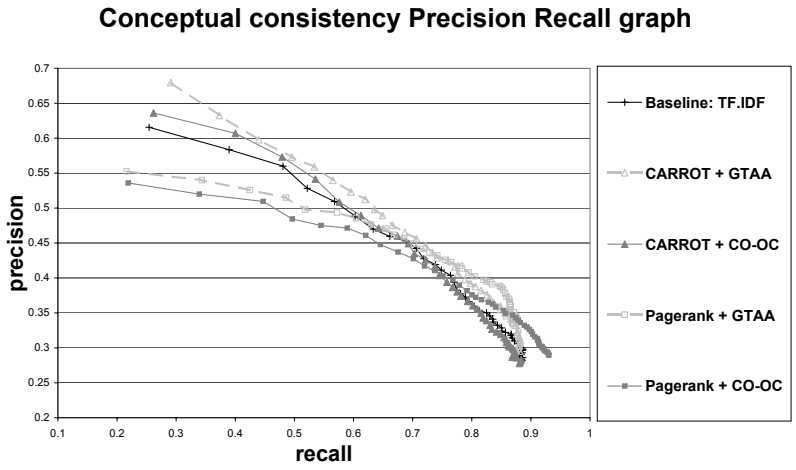


Fig. 1. Precision-recall graph showing conceptual consistency for the five different settings

3 Discussion and perspectives

The results show that it is possible to improve on TF.IDF based ranking by exploiting background knowledge. The simple rule based algorithm called CARROT improves upon the TF.IDF baseline for both types of background knowledge. Our pagerank-based algorithm however, did worse than the TF.IDF baseline. Although unexpected, it is quite logical: Pagerank, which only depends on the graph structure, cannot incorporate any frequency information.

The results also showed that the type of background knowledge is of influence: the thesaurus gave better results than the co-occurrences. The co-occurrences include too many random relations which unfortunately introduces noise. Taking the Mutual Information or Conditional Probabilities into account may give better results. This heuristic will be investigated in future work.

References

1. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th Anniversary Meeting of the ACL. (2002)
2. Wartena, C., Brussee, R., Gazendam, L., Huijsen, W.: Apolda: A practical tool for semantic annotation. The 4th International Workshop on Text-based Information Retrieval(TIR) (2007)
3. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems **30**(1–7) (1998) pp. 107–117
4. Iivonen, M.: Consistency in the selection of search concepts and search terms. Information Processing and Management **31**(2) (1995) pp. 173–190

Collaborative enterprise integrated modelling*

Chiara Ghidini[1], Marco Rospocher[1], Luciano Serafini[1], Andreas Faatz[2], Barbara Kump[3], Tobias Ley[4], Viktoria Pammer[3], and Stefanie Lindstaedt[4]

[1] FBK-irst, Via Sommarive 18 Povo, 38050,Trento, Italy
{ghidini,rospocher,serafini}@fbk.eu
[2] SAP Research, SAP AG. Bleichstraße 8, 64283 Darmstadt, Germany
andreas.faatz@sap.com
[3] Knowledge Management Institute, TU Graz. Inffeldgasse 21a, 8010 Graz, Austria
{bkump,viktoria.pammer}@tugraz.at
[4] Know-Center, Inffeldgasse 21a, 8010 Graz, Austria.
{tley,slind}@know-center.at

1 Introduction and Motivations

Enterprise modelling refers to the creation of an (integrated) enterprise model, that is, the structured description of one or more aspects of an enterprise and their mutual relations. Traditionally, enterprise models were focused on the description of process and business domain of an enterprise. Recently, enterprise modelling has been extended to other important assets of an enterprise (e.g., goals, human resources, enterprise structure and roles). Focusing on many different aspects of an enterprise (each one requiring specific modelling skills), and involving different modelling actors, enterprise modelling is truly a collaborative activity carried on under some collaborative protocol.

State-of-the-art methodologies and tools are mainly based on the, so called, *waterfall* paradigm. This paradigm presents some drawbacks towards an integrated enterprise modelling. First, the collaboration pattern has to stick to rigid interaction protocols which usually go from informal knowledge to formal knowledge. Second, the final formal model is an artefact which is not tightly integrated with the informal specifications that it is supposed to represent. These drawbacks greatly limit a real collaborative modelling between knowledge experts and knowledge engineers. A further limitation of many current methodologies and tools is that they usually deal with a single aspect of an enterprise. Not enough attention is given to the production of a reference meta-model for integrated enterprise models and to methodologies and tools for the support of a uniform integrated enterprise modelling.

Our work aims at supporting collaborative modelling of enterprises in two different ways. First, we propose a new collaborative approach for enterprise modelling, where different actors can actively collaborate in a truly flexible manner to create an integrated enterprise meta-model[5]. Second, we propose a tool based on Semantic MediaWiKi to support the development of an integrated enterprise meta-model. Please, see [1] for an extended version of this work, including a detailed related work section.

* Work partially funded under grant 027023. IST work programme of the European Community.
[5] This model was devised to support the development of work integrated learning applications and integrates a domain specific model, a process model and a competency model.

2 The Collaborative Enterprise Modelling Approach

The approach for enterprise modelling that we propose is inspired by recent Web 2.0 collaborative solutions, in particular wikis. In our approach all the different actors involved in modelling asynchronously collaborate towards the construction of an integrated enterprise model by inserting knowledge (either formal or informal), by transforming knowledge (from informal to formal) and by revising knowledge. A knowledge expert can enter knowledge - in form of informal knowledge - into the models, or provide feedback on the current models. The result of this input is stored in the "informal part" of the model. The system semi-automatically translates part of the informal model into a formal specification and vice-versa. Asynchronously, the knowledge engineer can refine the "formal part" of the model by inserting new statements and adding new constraints.

The result of the activity carried on under the collaborative enterprise modelling approach is the construction of strucure in which the different aspects of an enterprise are integrated in a unique model and in which a tight connection between the informal and formal part is retained. This integrated model is therefore an artefact that can be used both by humans and machines. The structure of this integrated model (hereafter called meta-model) is depicted in Figure 1. The main characteristic of this meta-model

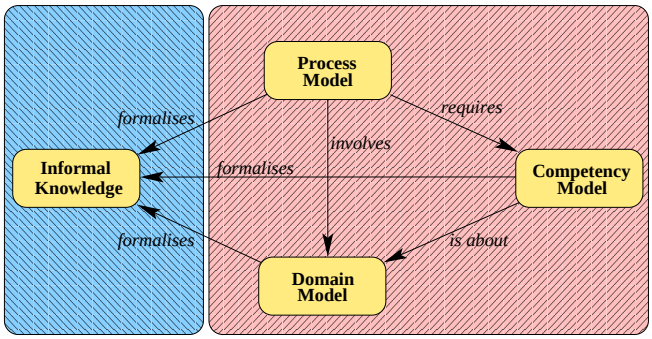


Fig. 1. The integrated enterprise model.

is the fact that it is structured in two components: the first component is the formal representation of the domain, the processes, and the competencies of an enterprise. These three aspects are described in three formal models, namely the *domain model*, the *process model*, and the *competency model*, which are bounded in a coherent integrated model. The second component is the *informal knowledge*. This component, which is usually left out of modelling schemata, contains an informal description of the formal model and it is tightly connected and intermixed with the elements of the formal model. We have decided to include also this part in the enterprise model as it has a crucial role in allowing human access and understanding of the integrated model. In our work, we have decide to represent the formal part of the meta-model as an OWL ontology, and the informal part as pages in a Semantic MediaWiki [2].

3 MoKi: the Modelling WiKi

To support the collaborative enterprise modelling here proposed we developed *MoKi*, the *Mo*delling Wi*Ki*, a tool based on Semantic MediaWiki. The main idea is to associate a wiki page to each (simple or complex) element of the formal model in a way that this page contains an informal but structured description of the element itself. The typical page contains (i) an informal description of the element considered, described mainly in natural language (images or drawings can be used as well), and (ii) a structured part, where the element is described by means of triplets of the form *(subject, relation, ojbect)*, with the element itself playing the role of the subject. This natural language based, but also structured, description provides an ideal bridge between formal and informal representation of knowledge.

To support the development of the integrated enterprise model, MoKi aims to offer a bunch of features to support the automatic alignment between the informal and formal knowledge coexisting in the models, and to ease the modelling of the three components (domain-specific model, process model, and competencies model) in a synchronised manner.

Here we briefly describe the features currently available in MoKi. The users can easily edit the content of wiki page by means of forms. Via the model import functionality some preexisting formal models can be imported in the wiki. Also list of elements organized according to predefined semantic structures (e.g. a taxonomy or a mereology) can be easily imported. MoKi includes a term extraction functionality which allows to add to the models terms (or clusters of terms) extracted from digital documents. Browsing/editing of the models is supported by means of a graphical interface. The informal models described in MoKi can be easily exported in the appropriate formal language thanks to the model export functionality.

4 Conclusions

In this paper we have presented a (i) new collaborative approach for enterprise modelling, and (ii) a wiki-based tool to support it (MoKi). The approach and the tool have been successfully applied within EU-project APOSDLE (`www.aposdle.org`) to develop five integrated enterprise models in the following domains: environmental consultancy, electromagnetism simulation, innovation and knowledge management, requirements engineering, and statistical data analysis.

References

1. Ghidini, C., Rospocher, M., Serafini, L., Faatz, A., Kump, B., Ley, T., Pammer, V., Lindstaedt, S.: Collaborative enterprise integrated modelling. Technical Report 200806005, FBK-irst (2008)
2. Schaffert, S., Gruber, A., Westenthaler, R.: A semantic wiki for collaborative knowledge formation. In: Proceedings of SEMANTICS 2005 Conference., Vienna, Austria (2005)

NeOn Methodology: Scenarios for Building Networks of Ontologies

Asunción Gómez-Pérez and Mari Carmen Suárez-Figueroa
Ontology Engineering Group. Departamento de Inteligencia Artificial.
Facultad de Informática. Universidad Politécnica de Madrid
{asun, mcsuarez}@fi.upm.es

Abstract. In this poster we present the NeOn methodology[1] that identifies *nine scenarios for building ontology networks* collaboratively, emphasizing the reuse and the reengineering of ontological and non ontological resources.

Keywords: ontology engineering, ontology development, reuse, reengineering.

1. Introduction

The 1990s and the first years of this new millennium have witnessed a growing interest of many practitioners in methodologies (e.g., METHONTOLOGY, On-To-Knowledge, DILIGENT, etc.) that support the creation of ontologies from scratch. All these approaches have transformed the art of constructing ontologies into an engineering activity. A series of existing methodologies have been reported in [1].

With the goal of speeding up the ontology development process, ontology practitioners are starting to reuse and reengineer knowledge-aware resources, which have already reached some degree of consensus. The SEEMP[2] project, for example, includes ontologies that were developed by reusing and reengineering existing human resources management standards. The development of the SEEMP ontologies reveals that current methodologies are very rigid and do not cover complex scenarios in which the reuse and reengineering of knowledge resources are considered.

The NeOn project[3] foresees that the Semantic Web of the future will be characterized by using a very large number of ontologies embedded in ontology networks[4] built collaboratively by distributed teams and coming from different sources. Such networks could include ontologies that already exist or that could be developed by reusing other ontologies and/or non ontological but knowledge-aware resources (thesauri, lexicons, databases, UML diagrams, etc.). To achieve this goal, the NeOn project intends to create the *NeOn methodology*, a methodology that

[1] This work has been supported by the NeOn project (IST-2005-027595)
[2] http://www.seemp.org/
[3] http://www.neon-project.org/
[4] An ontology network or a network of ontologies is defined as a collection of ontologies related together through a variety of different relationships such as mapping, modularization, version, and dependency relationships [3]

supports the collaborative aspects of ontology development as well as the reuse and the dynamic evolution of networked ontologies in distributed environments.

Thus, it is not premature to affirm that a new ontology development paradigm is starting, whose emphasis is on the reuse and possible subsequent reengineering of knowledge-aware resources, on the collaborative and argumentative ontology development, and on the building of ontology networks.

In this poster we present a set of *nine scenarios for building ontologies and ontology networks* collaboratively emphasizing the reuse and reengineering.

2. NeOn Scenarios for Building Ontology Networks

Fig. 1 presents the set of the 9 most plausible scenarios for building ontologies and ontology networks. The directed arrows with associated numbered circles represent the different scenarios. Each scenario is decomposed into different processes or activities. Processes and activities are represented with colored circles or with rounded boxes, and are defined in the NeOn Glossary [5]. Fig. 1 also shows (as dotted boxes) the existing knowledge resources to be reused, and the possible outputs that result from the execution of some of the presented scenarios.

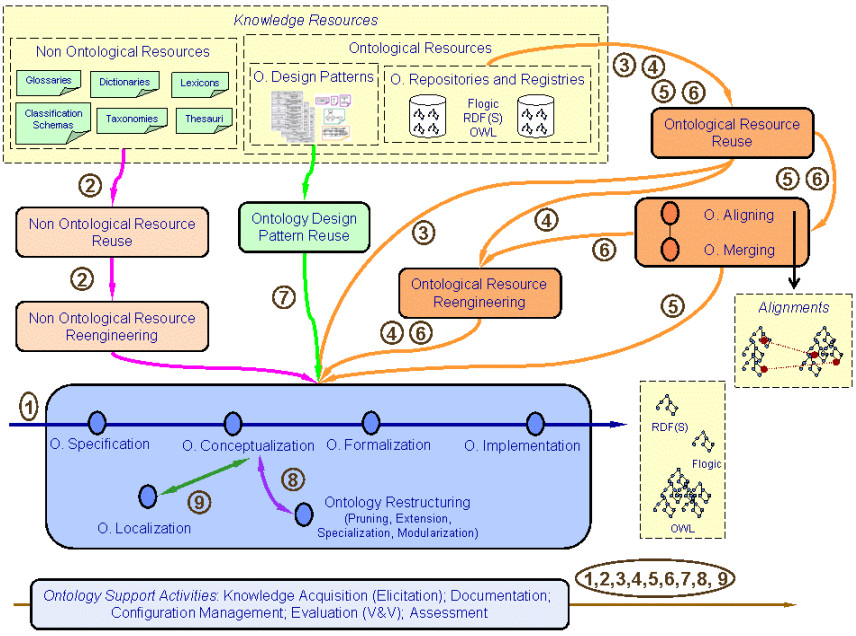


Fig. 1. Scenarios for Building Ontologies and Ontology Networks

- *Scenario 1*: From specification to implementation.
- *Scenario 2*: Reusing and reengineering non ontological resources.
- *Scenario 3*: Reusing ontological resources.
- *Scenario 4*: Reusing and reengineering ontological resources.
- *Scenario 5*: Reusing and merging ontological resources.

- *Scenario 6*: Reusing, merging and reengineering ontological resources.
- *Scenario 7*: Reusing ontology design patterns.
- *Scenario 8*: Restructuring ontological resources.
- *Scenario 9*: Localizing ontological resources.

Knowledge acquisition, documentation, configuration management, evaluation and assessment should be carried out all along the ontology development.

From this set of scenarios, we can say that scenario 1 is the most typical for building ontologies and ontology networks without reusing existing knowledge resources. Moreover, the identified scenarios within the NeOn methodology are flexible since their combination is allowed within the development of ontologies and ontology networks. It is worth mentioning that any combination of scenarios should include scenario 1, since this scenario is made up of the core activities that have to be performed in any ontology development. Indeed, as Fig. 1 shows, the results of any other scenario should be integrated in the corresponding activity of scenario 1.

To date, the first version of the NeOn methodology [4] includes guidelines for processes and activities of scenarios 1, 2, 3 and 7. On the other hand, this methodology is being evaluated within the development of the ontologies in two NeOn use cases: invoice management and semantic nomenclature [2].

3. Conclusion

The NeOn methodology has identified a set of nine flexible scenarios for collaboratively building ontologies and ontology networks, with special emphasis on reusing and reengineering knowledge-aware resources (ontological and non ontological). Unlike the rigid scenario for building ontologies presented in METHONTOLOGY, On-To-Knowledge and DILIGENT - a scenario that ranges from the specification to the implementation-, the scenarios here proposed are flexible because the NeOn methodology permits their combination for building ontologies.

References

1. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: *Ontological Engineering*. November 2003. Springer Verlag. Advanced Information and Knowledge Processing series. ISBN 1-85233-551-3.
2. Gómez-Pérez, J.M., Pariente, T., Buil-Aranda, C., Herrero, G., Baena, A.: *NeOn Deliverable D8.3.1. Ontologies for pharmaceutical case studies*. NeOn project. http://www.neon-project.org. 2007.
3. Haase, P., Rudolph, S., Wang, Y., Brockmans, S., Palma, R., Euzenat, J., d'Aquin, M.: *NeOn Deliverable D1.1.1 Networked Ontology Model*. November 2006.
4. Suárez-Figueroa, M.C., Dellschaft, K., Montiel-Ponsoda, E., Villazon-Terrazas, B., Yufei, Z., Aguado de Cea, G., García, A., Fernández-López, M., Gómez-Pérez, A., Espinoza, M., Sabou, M.: *NeOn D5.4.1. NeOn Methodology for Building Contextualized Ontology Networks*. NeOn project. http://www.neon-project.org. February 2008.
5. Suárez-Figueroa, M.C., Gómez-Pérez, A.: *Towards a Glossary of Activities in the Ontology Engineering Field*. 6th Language Resources and Evaluation Conference (LREC 2008). Marrakech (Morocco). May 2008.

Problem Solving Methods as Semantic Overlays for Provenance Analysis

Jose Manuel Gómez-Pérez[1] and Oscar Corcho[2]

[1] iSOCO S.A., Pedro de Valdivia 10. 28006 Madrid, Spain
jmgomez@isoco.com
[2] Ontology Engineering Group. Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid. 28660 Boadilla del Monte, Madrid, Spain
ocorcho@fi.upm.es

Abstract. Processes executed in data-intensive domains produce large amounts of provenance information. Thus, sophisticated analytical capabilities with a higher level of abstraction are required that provide users with meaningful interpretations of process executions, explaining provenance in a way closer to how domain experts reason on a given problem and facilitating their comprehension. In this work, we use Problem Solving Methods as semantic overlays that, sitting on top of process documentation, provide domain experts with meaningful interpretations of provenance.

1 Introduction: Towards Knowledge Provenance

Provenance is broadly defined as the origin or source from which something comes, and the history of subsequent owners. In the context of data, process and computation-intensive disciplines, such as physics, biology, astronomy, etc., to name but a few, provenance is focused on the description and understanding of where and how data is produced, the actors involved in the production of such data, and the processes applied to the object before arriving in the collection from which it is now being accessed, so that it can be considered as an important source of information to determine its overall quality. In a usual discovery task, scientists integrate data from data sources, filter the combined data according to some criteria, and annotate the data with information about the relationships that have just been discovered. All the tasks applied in this process contribute to the provenance record of that data product.

According to [5], provenance information can be seen as a pyramid with four main levels: Data, Organization, Process, and Knowledge. Most of the current provenance systems are focused on the first three levels, providing means for recording and querying process documentation. Our approach focuses on the upper level (knowledge), by applying Problem Solving Methods [1] (PSMs) as semantic overlays that provide a meaningful interpretation of such information. This approach emphasizes the role of PSMs as reusable and generic strategies [2] for modelling and reasoning with problem-solving behaviour at the knowledge level. We aim to support the interpretation of provenance by subject-matter experts (SMEs) with little background in computer science.

SMEs are assisted in two different ways. First, the semantic overlay provided by PSMs can be used by SMEs as an abstract specification of the process. The execution of the process can be validated against such specification by means of its provenance. SMEs can relate, manually or automatically, the services contained within the process with the PSM and then query the process provenance log in terms of the higher level of abstraction provided by these overlays, instead of the low-level operations stored by the provenance log. Second, processes defined by SMEs can be complex and use the different data sources in many varied ways until the desired results are obtained. By means of determining which one from amidst the available PSMs in the PSM library provide a better description of the manipulation of these data sources, SMEs can get a better grasp of the process that has been executed and assimilate it.

Thus, we use PSMs as semantic overlays that allow representing provenance in terms of the domain, at multiple levels of abstraction, accomplishing a threefold goal: i) to facilitate users the understanding of how provenance information relates with the execution of their processes, ii) to simplify the analysis of process executions by showing their decomposition into domain-level subprocesses, and iii) to visualize the execution of a process at different levels of detail.

2 KOPE: A Knowledge-Oriented Provenance Environment

Our approach to knowledge provenance is implemented as the Knowledge-Oriented Provenance Environment (KOPE). KOPE requires the following knowledge resources: i) a *PSM metamodel* describing PSM constructs and how they are related with each other, ii) a *PSM library* containing a hierarchy of methods, instances of the PSM metamodel, and iii) *domain ontologies*, describing the application domain.

The KOPE architecture (Figure 1) is built by three main building blocks: an underlying provenance infrastructure based on PASOA [3], providing functionalities for documenting process execution and querying of this information from the provenance store, a PSM editor that allows managing PSM libraries and domain ontologies as well as visualizing provenance information at multiple levels of detail, and the KOPE engine, which uses the methods contained in the PSM libraries and the ontologies modelling the domain to analyze process executions.

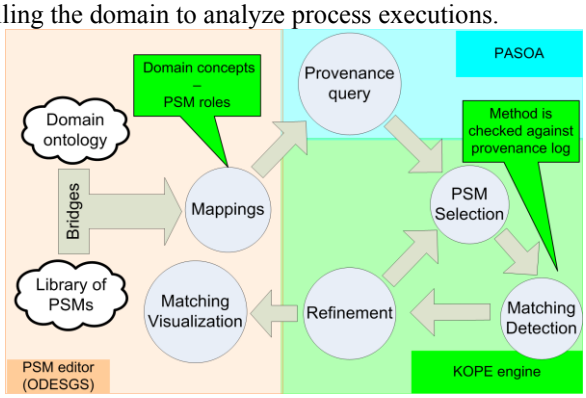


Figure 1: Overall KOPE architecture

We extend process documentation with semantic annotations, in terms of domain ontologies, of the data exchanged between interacting services. Such metadata are automatically produced during process execution by the actors participating in the process, as part of the process documentation. We use interaction p-assertions, which document service message exchange at the application level, specifically the *content* tag, as the carrier of these semantic metadata. Since domain and PSM entities are related by means of *bridges*, such metadata allow analyzing provenance in terms of the domain (e.g. as in [4]) according to the generic descriptions of the processes as provided by the methods of the PSM library.

The KOPE engine matches, at each decomposition level provided by a PSM, the knowledge flow of such PSM against the PASOA documentation of a process execution, which follows a directed acyclic graph structure (p-DAG) formed by interaction and relationship p-assertions. The goal of the algorithm implemented in the engine is to detect whether the twigs[1] between inputs and outputs of the knowledge flow of the PSM occur as well in the p-DAG of the process execution and, consequently, the p-DAG (and therefore, the process execution it represents) can be considered as an occurrence of the PSM in a particular domain of application.

KOPE has been evaluated in the context of the Provenance Challenge[2]. KOPE participated in this challenge with a twofold goal: i) to evaluate interoperation with other provenance systems, in particular with PASOA, whose infrastructure and data model support process documentation in the KOPE architecture and ii) to evaluate its capabilities for analyzing provenance information at the knowledge level, at multiple levels of abstraction and detail, employing PSMs as a novel paradigm for knowledge provenance.

Acknowledgements

This work has been funded as part of the IST-2005-027595 EU project NeOn, IST-2007-215040 EU project ACTIVE, and IST-2007-215219 EU project SOA4ALL.

References

1 McDermott, J. Preliminary steps towards a taxonomy of problem-solving methods. In Marcus, S., editor, Automating Knowledge Acquisition for Expert Systems, pages 225-255. Boston, Kluwer.

2 Motta, E. (1999). Reusable Components for Knowledge Modelling. IOS Press, Amsterdam. November 1999.

3 Munroe, S., Groth, P., Jiang, S., Miles, S., Tan, V., Moreau, L. Data model for Process Documentation. Technical report, University of Southampton, 2006.

4 Wong, S. C., Miles, S., Fang, W., Groth, P. and Moreau, L. (2005) Provenance-based Validation of E-Science Experiments, in Gil, Y., Motta, E., Benjamins, V. R. and Musen, M. A., Eds. *Proceedings of 4th International Semantic Web Conference (ISWC), Lecture Notes in Computer Science vol 3729*, pp. 801-815. Springer-Verlag.

5 Zhao, J., Wroe, C., Goble, C., Stevens, R., Quan, D., Greenwood, M. Using Semantic Web Technologies for Representing e-Science Provenance Proc 3rd International Semantic Web Conference ISWC2004, Hiroshima, Japan, 9-11 Nov 2004, Springer LNCS Hiroshima, Japan, 2004.

[1] As opposed to paths, twigs are multibranched connections between nodes in a graph.

[2] twiki.ipaw.info/bin/view/Challenge/WebHome

Collaboration Patterns in a Medical Community of Practice*

Marie Gustafsson[1,2], Göran Falkman[1], Olof Torgersson[2], and Mats Jontell[3]

[1] School of Humanities and Informatics,
University of Skövde, SE-541 28 Skövde, Sweden
{marie.gustafsson,goran.falkman}@his.se
[2] Department of Computer Science and Engineering, Chalmers
University of Technology/Göteborg University, SE-412 96 Göteborg, Sweden
{mariegus,oloft}@chalmers.se
[3] Institute of Ontology, Sahlgrenska Academy, Göteborg University,
P.O. Box 450, SE-405 30 Göteborg, Sweden
Mats.Jontell@odontologi.gu.se

1 Introduction

Since the mid 1990's, the Swedish Oral Medicine Network (SOMNet) has promoted the harmonization and dissemination of knowledge and the sharing of clinical experience within oral medicine. Its members are located throughout Sweden and are mainly dentists with a professional interest in oral medicine. SOMNet holds monthly teleconference meetings focused on case consultations. An assigned chairperson leads the meeting, guides case presentations, sums up discussions, and records decisions made. When presenting a case, the presenter "tells the story" of his/her encounters with the patient and reports on treatments tried and results achieved so far. Then, the other participants ask questions of clarification and start suggesting possible diagnosis and treatments. Similar cases or general treatment strategies will sometimes accompany the suggestions.

SOMWeb is an online system supporting SOMNet's activities by providing facilities for adding and administering cases to be discussed at SOMNet meetings; browsing cases, meetings, and members of the system; looking at presentations of individual cases and meetings; administering meetings; and reading news. As described in previous work [1], community aspects (e.g., users, meetings, cases, and templates) of SOMWeb are modeled in OWL and data is stored as RDF. The SOMWeb system was introduced in May 2006 and by April 2008, SOMWeb has 90 users, 89 cases have been added, and 20 meetings have utilized SOMWeb.

In our previous research, we have studied clinicians' use of SOMWeb [3] as well as the possibility of using ideas from the Pragmatic Web [4] to describe communications patterns within the community [5]. In this paper, we continue this research by presenting ideas on how collaboration patterns within the domain can be identified, modeled, and be put into use.

* This work is funded by the Swedish Governmental Agency for Innovation Systems (VINNOVA), research grant 2006-02792.

2 Collaboration Patterns

In identifying patterns of collaboration, we studied SOMNet and their use of the SOMWeb system by observing ten meetings, interviewing nine members, and using an online questionnaire. See [3] for details of these studies. Literature also informed the construction of patterns, e.g., [6], [7], and [8]. Particularly, we view SOMNet as a community of practice (CoP), a group of people sharing "a concern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in this area by interacting on an ongoing basis" [7].

Patterns identified are the request pattern, activity pattern, and case activity pattern. The request pattern initiates an activity pattern, and in general this is achieved by a member entering a case into SOMWeb and requesting input on it (usually by putting it up for discussion at a meeting). The activity pattern describes a general activity of SOMNet. Case activities are activities which take requests about cases as input. There are three subclasses of case activity: Case consultation activity, case discussion activity, and case sharing activity.

Our collaboration patterns are composed of classes from our ontologies for users, organizations, and oral medicine. In the user ontology we have used the idea of [7] that members of a CoP have different levels of participation, where they can be divided into groups of core, active, and peripheral members. We have observed a similar division within SOMNet. Currently, a `CoreMember` is defined as a member with oral medicine certification or who has chaired at least one meeting. An `ActiveMember` is defined as a member that has added at least one case. A `PeripheralMember` is a member who is not entailed by the definition of `CoreMember` or `ActiveMember`.

We here go into detail on the case consultation activity, as depicted in Fig. 1. In this activity, members participate in different roles, one of which is a moderator, who is defined to be a `NonNoviceMember`, which is further defined as a member who is a `CoreMember` or `ActiveMember`. A `CaseConsultation` results in a `Decision`. The `Decision` class has several subclasses such as `DiagnosisDecision`,

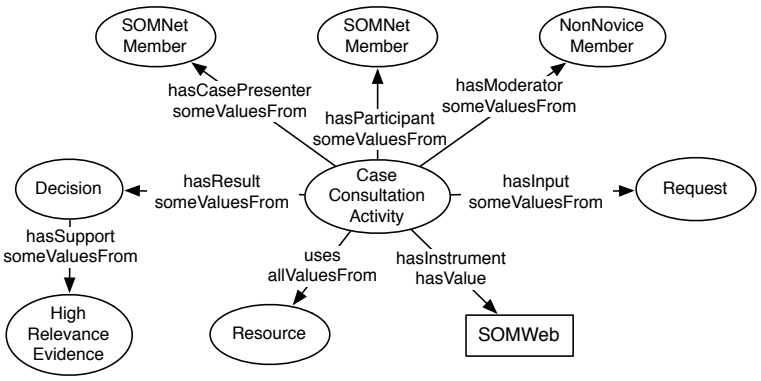


Fig. 1. The consultation pattern describes a specialization of an activity.

`FollowUpDecision`, `TreatmentPlanDecision`, and `GeneralAdviceDecision`. A `Decision` is supported by `Evidence`, a subclass of `Support`, whose relevance can be high, medium, low, or none.

The presented collaboration patterns can be used to guide the system and its users, though this is not yet implemented. For example, the kind of case consultation pattern invoked could influence the amount of detail needed in the case description. The pattern can also be used in the assigning of chairpersons to meetings and as guidance to that chairperson. After a meeting, the system could use the pattern to ensure that a decision has been recorded and that this decision is supported by evidence of adequate relevance.

3 Discussion

The objective of our research is to better understand collaboration and interaction between clinicians, in order to improve IT tools that support evidence-based medicine. In the short term, this translates to elaborate the presented patterns, perhaps taking additional pattern theories and models into account (e.g., social networking and narratives). Also, we want to generalize from our experience methods for elucidating collaboration patterns and provide experience of putting the patterns to use in IT tools. In the longer term, since co-operative care is a fundamental part of evidence-based care in *any* medical discipline, developing SOMWeb into a general tool that builds online CoPs for other disciplines from a pattern-based description of the domain in question is an interesting prospect.

References

1. Gustafsson, M., Falkman, G., Lindahl, F., Torgersson, O.: Enabling an online community for sharing oral medicine cases using Semantic web technologies. In: Proc. 5th Int. Semantic Web Conference (ISWC 2006). Volume 4273 of Lect. Notes Comput. Sci. (2006) 820–832
2. Wenger, E.: Communities of practice: Learning, meaning, and identity. Cambridge University Press, Cambridge, U.K. and New York, N.Y. (1998)
3. Falkman, G., Gustafsson, M., Jontell, M., Torgersson, O.: SOMWeb: A Semantic Web-based system for supporting collaboration of distributed medical communities of practice. J. Med. Internet Res. (Medicine 2.0) (2008) Forthcoming
4. Schoop, M., de Moor, A., Dietz, J.L.: The Pragmatic web: A manifesto. Commun. ACM **49**(5) (2006) 75–76
5. Falkman, G., Gustafsson, M., Torgersson, O., Jontell, M.: Towards pragmatic patterns for clinical knowledge management. In: Proc. 2nd Int. Conference on the Pragmatic Web (ICPW'07), ACM (2007) 65–74
6. de Moor, A.: Community memory activation with collaboration patterns. In: Proc. 3rd Conf. on Community Informatics Research Network (CIRN 2006). (2006) 1–18
7. Wenger, E., McDermott, R., Snyder, W.: Cultivating Communities of Practice. Harvard Business School Press, Boston, MA (2002)
8. Kane, B., Luz, S.: Multidisciplinary medical team meetings: An analysis of collaborative working with special attention to timing and teleconferencing. Comput. Support. Coop. Work **15**(5–6) (2006) 501–535

iMERGE: Interactive Ontology Merging

Zoulfa El Jerroudi and Jürgen Ziegler

University Duisburg-Essen, 47057 Duisburg, Germany

Abstract. In this paper we present novel visual analytics techniques which help the user in the process of interactive ontology mapping and merging. A major contribution will be the strong integration and coupling of interactive visualizations with the merging process enabling the user to follow why concepts are merged and at which position in the ontology they are merged. For this purpose, adapted ontology similarity measures and new techniques for representing ontologies will be required to enable responsive, real time visualization and exploration of the comparing and merging results.

1 Visual Ontology Mapping

The areas of ontology mapping and ontology merging have largely relied on automatic and semi-automatic methods in the past (FOAM [1], PROMPT [2], OLA [3] and FCA-Merge [4]), where user control and interaction is limited and results are typically only presented to the user at the end of some complex computational process. The effectiveness of these approaches can be increased in many application domains, if these approaches use the users knowledge and expertise in the comparing and merging process and support the explorative and iterative activities that are essential for the user's sensemaking process. Ontology merging is still largely a human-mediated process. The user could not trust in automatic results, where he does not know where and for which reason concepts are merged.

In this paper we present an approach, which helps to enable users to explore the ontology and to compare results in an intuitive and efficient manner. We aim to support the analytic comparing and merging process providing tightly linked and integrated techniques and views for visualizing and exploring the raw ontologies and derived merging results. For this purpose we develop a prototype editor iMerge. The introduced editor iMerge provides different views for this purpose(see Fig. 1).

The *SmartTree-View* [5] extends the conventional tree widget with a number of mechanisms facilitating ontology exploration and development. In addition to the hierarchical structure shown (typically the class hierarchy), non-hierarchical relations are shown dynamically upon selecting a node. SmartTree introduces *Condense+Expand* and *Prune+Grow*, two new interaction techniques allowing to hide and expose parts of the tree.

The *Matrix-View* [6] is suited for comparing two ontologies and determining where most of the mappings between ontologies occur. In the *Matrix-View* the

2

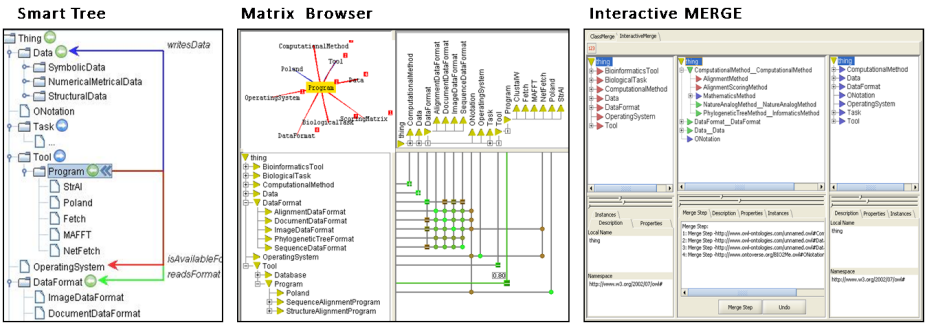


Fig. 1: Different Views of the ontology and the results of comparison

ontologies to be compared are confronted on the both axes of the matrix. High agreement in the ontologies are signified with green symbols at junctures, parts which are different are signalised with red symbols. A plus symbol in the matrix indicates that there are similar concepts hidden in the substructure. For the comparing process different algorithms can be selected by the user. Based on the results of comparison the ontologies can be merged.

The *InteractiveMERGE-View* supports merging of two ontology step by step and with leverage the users knowledge and expertise. For supporting this task, both ontologies are highlighted in different colors, so the user can register from which parts the changes comes from. The differences are shown first in the original ontology to the user and after that the consequences of the merging step are shown visually in the target ontology. The domain expert can *accept, change* or even *reject* this step. The alternative views ease the ontology designers to comprehend the consequences of their work.

Similarity measures for ontological structures have been widely researched, e.g. in [1], [2], [3] and [4]. A survey of ontology mapping is given in Falconer [7] and Choi [8]. The mentioned approaches do not give attention to the interactive aspects of ontology mapping and merging (except PROMPT [2]). Falconer et. al. [9] have defined requirements which should support the user in the cognitive tasks for ontology mapping and merging. The proposed iMerge editor considers the following important requirements:

- Support ontology exploration and manual creation of mappings
- Provide a visual representation of the source and target ontology
- Provide a method for the user to accept/reject a suggested mapping
- Provide access to full definitions of ontology terms
- Show the context of a term when a user is inspecting a suggested mapping
- Provide interactive access to source and target ontologies
- Support interactive navigation and allow the user to accept/reject mappings
- Provide progress feedback on the overall mapping process

Only the automatic verification of the supposed mapping is not done by the proposed editor. The iMerge proposes some mappings but does not consider possible conflicts which may occur if the concepts are merged.

2 Components of the iMerge-Editor

The structure of the proposed editor is divided in the units shown in Fig. 2.
Visualisation: In the first step of the visual ontology exploration the user needs

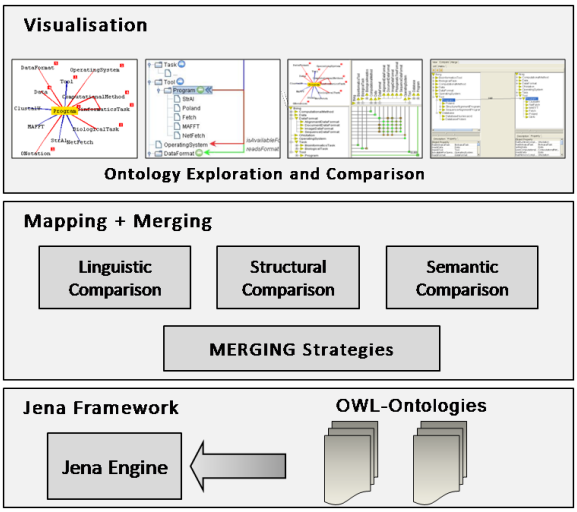


Fig. 2: Units of the iMerge editor

to get an overview of the ontologies, which have to be merged. Here, the user needs different access points (SmartTree, Graph-Viz) to the ontology, because he sets his exploration objectives in the most time during the interaction with and navigation in the ontology. The different views should be coupled in a way, that even if changing the view, the actual focus remains clear. After the visual exploration, the focus switches to the proposed mappings.

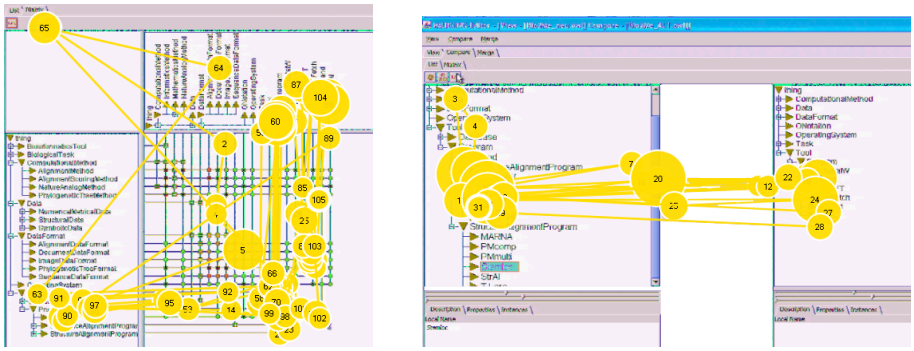
Mapping + Merging: This component provides methods for identifying mappings between the source and target ontology, which try to approximate the understanding of what the users consider to be a good match. For this purpose, our approach combines the results of several independently executed match algorithms.
The **linguistic approach** exploits text-based properties of the ontologies, such as name and description. With the method *EditDistance* [10] string similarity is computed from the number of edit operations (insertions, deletions, substitutions of single characters) necessary to transform one string into another one. As

an alternative the method *N-Gram* [11] can be used. Here, strings are compared according to their set of n-grams, i.e., sequences of n characters.
The **structural approach** exploits relationships between concepts that appear together in a structure. Usually, concepts and their relations are represented in a graph so that different kinds of structural related elements can be identified for matching. To estimate the similarity between two concepts, we can compare different kinds of their neighbor elements, such as the *parents, children*, or the *leaves* subsumed by them.
The **semantic approach** estimates the similarity between concepts based on their terminological relationships, such as synonymy, hypernymy, hyponymy. This approach requires the use of auxiliary sources, such as documents or annotations, in which the semantic relationship is captured. This method takes as input two ontologies and a set of documents which are linked with the concepts. We assume that, if documents annotated with concept a (of O_1) are similar to the documents annotated with concept b (of O_2), then the concepts a and b are similar.
Merging Strategy: This component develops step by step a new ontology G based on the preceding comparison, where the user can follow why the concepts are merged. First, concepts without a mapping pair are copied in the resulting ontology G. Concepts with a mapping candidate should be merged. For merging two concepts the user has to specify a threshold. Similar sub-concepts and properties with a similarity value higher than the threshold are merged recursively. The color indicates if a concept comes from O_1 or O_2.

3 Discussion and Further Work

Within this work informal usability tests are conducted, which give first hints for the correctness of the assumptions*(Users could not trust in automatically generated merging results, because they could not follow why concepts are merged)*. Furthermore, the visual exploration of the mapping pairs has been compared with the eye tracking system Tobii T60 both in the *Matrix View* and the *List View* (see Fig. 3). The results of the eye tracking analysis confirm the assumption, that both tasks *get an overview* and *comprehend details* need different views. In the *List View* the fixations are concentrated only on the two concepts that are compared and the gaze motion goes between these both concepts. In the *Matrix View*, not only the comparing concepts are regarded, but also nearly concepts are considered. The defined matrix leads to consider concepts in the neighborhood. It is also visible, that in the *Matrix View* the number of fixations in the same time is higher, but the duration is smaller. In this view the user tries to get only a fast overview about the ontology.

Based on these results further work goes in coupling different views with the merging process more directly, e.g. conflicts during the merging process can be shown visually. Furthermore, we should consider the development and changes in the ontology over time especially for different ontology versions.



(a) Matrix View (b) List View

Fig. 3: Gaze Motion: *Matrix View* and *List View*

References

1. Ehrig, M.: Ontology Alignment: Bridging the Semantic Gap (Semantic Web and Beyond). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
2. Noy, N.F., Musen, M.A.: The prompt suite: interactive tools for ontology merging and mapping. Int. J. Hum.-Comput. Stud. **59**(6) (2003) 983–1024
3. Euzenat, J., Valtchev, P.: Similarity-based ontology alignment in owl-lite. In de Mántaras, R.L., Saitta, L., eds.: Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-04), IOS Press (2004) 333–337
4. Stumme, G., Maedche, A.: FCA-MERGE: Bottom-Up Merging of Ontologies. In: Proc. of the 17^{th} International Joint Conference on Artificial Intelligence (IJCAI). (2001) 225–234
5. Hüsken, P., Ziegler, J.: Degree-of-interest visualization for ontology exploration. In Baranauskas, M.C.C., Palanque, P.A., Abascal, J., Barbosa, S.D.J., eds.: INTERACT (1). Volume 4662 of Lecture Notes in Computer Science., Springer (2007) 116–119
6. Ziegler, J., Kunz, C., Botsch, V., Schneeberger, J.: Visualizing and exploring large networked information spaces with matrix browser. In: : Information Visualisation, 2002. Proceedings. Sixth International Conference on. (2002) 361–366
7. Falconer, S., Noy, N., Storey, M.A.: Ontology mapping - a user survey. In Shvaiko, P., Euzenat, J., Giunchiglia, F., He, B., eds.: Proceedings of the Workshop on Ontology Matching (OM2007) at ISWC/ASWC2007, Busan, South Korea. (November 2007)
8. Choi, N., Song, I.Y., Han, H.: A survey on ontology mapping. SIGMOD Rec. **35**(3) (2006) 34–41
9. Falconer, S.M., Storey, M.A.D.: A cognitive support framework for ontology mapping. In Aberer, K., Choi, K.S., Noy, N.F., et al., D.A., eds.: ISWC/ASWC. Volume 4825 of Lecture Notes in Computer Science., Springer (2007) 114–127
10. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8 (1966)
11. Damashek, M.: Gauging similarity with n-grams: Language-independent categorization of text. Science **267**(5199) (February 1995) 843–848

Semantic cartography: towards helping experts in their indexation task

Eric KERGOSIEN, Marie-Noelle BESSAGNET, Mauro GAIO
UPPA, Laboratory LIUPPA, 64000 PAU
{eric.kergosien, marie-noelle.bessagnet, mauro.gaio} @univ-pau.fr
http://liuppa.univ-pau.fr

Abstract: Our approach aims at helping experts in their indexation work using the relationship between concepts included in descriptive notices defined by using an external semantic structure (taxonomy, thesaurus, etc). In our research, we exploit specificities of the corpus linked to words which "have a meaning" to experts during the design of a descriptive notice. We propose tools in order to visualize the indexation work for validation by the experts.

Keywords: Knowledge Engineering, Thesaurus, Knowledge Representation

1 Introduction

Thanks to experts' indexation efforts, documents have rich descriptions, notably descriptive notices, described on a well-known semantic structure-base (taxonomy, thesaurus, etc). Our research focuses on a semi-automatic validation of the manual indexation work by exploiting expert knowledge (list of keywords chosen in the semantic structure-base) automatically extracted from notices. This approach has two steps: (i) describe information by expert knowledge which is automatically extracted from notices; (ii) give the possibility to navigate within the collection via the identified knowledge representing the indexation work. In a first part (&2), we expose our objectives. Thus, we can develop our approach (&3) to design a specific semantic structure for exploration in a corpus indexed by experts (librarians using RAMEAU[1] in our case). We make propositions (&4) to help the validation of the indexation work within the corpus based on the enriched thesaurus.

2 Objectives

With the aim of proposing a validation tool to experts for the use of controlled vocabularies[2] which they apply to fit their documents analysis, we develop two preliminary steps in our approach. The first step allows extracting and structuring knowledge of our corpus made up of descriptive notices and the controlled vocabulary used to describe these notices. In this step, we connect to research work such as [1] which sets out to mix terms from a thesaurus and terms from other sources

1 RAMEAU : http://rameau.bnf.fr/ is a french thesaurus defined by the Bibliothèque Nationale de France (BNF) for or the majority of the French libraries

2 A vocabulary is called "controlled" if it is defined with three levels of control: a semantic one, a terminological one and a syntactical one. We use the RAMEAU thesaurus. RAMEAU is the controlled vocabulary of indexation of numerous libraries

to better point information retrieval within management system. One of our objectives is the design of a process to transform a classic controlled vocabulary into a knowledge base [2]. The second step proposes a representation of this structure, based on concepts map [3]. In the end, we use techniques of semantic cartography [4] to tackle in a synthetic way the complete indexation work of a given collection.

3 Design and visualization of a domain-specific thesaurus

According to the AFNOR[3], a thesaurus is a documentary language based on a hierarchical structuring for one or more knowledge domains; notions are represented by terms from one or more natural language and relationships between notions by conventional signs. The first step automatically extracts "root-words" of the controlled vocabulary. In our experimentation, these root-words are selected by librarians within RAMEAU and used in XML descriptive notices (figure 1).

```
<DEE>Eaux minérales -- Barèges (Hautes-Pyrénées) -- 18e s.</DEE>
<TITRE>Observation sur les eaux minérales de Bigorre et du Béarn</TITRE>
<LEGENDE> T.de Bourdeu lance la mode du thermalisme pyrénéen</LEGENDE>
<DATE>2007-04-16</DATE>
```

Fig. 1 – Extract of descriptive notice 1

Each tag DEE contains several root-words separated by element "–". For each term found in a notice, we attach within the XML Topics Map structure [5] a link to the document. Root-words represent the conceptual level and document the physical level. This list of root-words is a first step towards the definition of a thesaurus depicting part of librarians' knowledge on the collection; it remains to identify set of relations between these terms in order to develop a sub-thesaurus. By exploiting the controlled vocabulary base structure, we automatically improve the above vocabulary with: "generic", "used for" and "related" (for thesaurus) terms; relations which are linked. Then, we have developed a tool which proposes a map representation allowing experts to tackle in a synthetic way the complete indexation work of a given collection, realized by different librarians. Up to now, it has been very difficult to represent due to the large number of documents and associated descriptive terms. In our tool, an interface gives a global view and a local view. The global view allows grasping the expert knowledge structured in the thesaurus in an integral way; the local view, based-on conceptual map representation highlights a subset of the semantic structure [6]. We explicitly represent the existing relationships between the terms of the thesaurus (conceptual level) and the documents of the collection related to these terms (physical level). The combination of both views facilitates navigation and thus the re-reading of the indexation work completed on the collection.

4 Suggestions for assistance in the validation of indexation work

The first feedback from experts of the media library is encouraging, showing the possibility to propose a synthetic display of their indexation work. Following the

3 AFNOR is a French institution for standardization. Documentation : règles d'établissement des thésaurus monolingues. NF Z47-100, 1981.

automatic thesaurus creation, an automatic control process begins to make sure that the terms used in the descriptive notices indexing the collection, are properly labelled as root-words in the given controlled vocabulary. This process identifies 4 types of errors due to the usage of the controlled vocabulary: *(i) Documents without descriptive notices; (ii) Incorrect information on root-words; (iii) Management of non-selected terms; (iv) Management of homonymy*. Concerning cases linked to name errors and missing words (i) (ii), certain solutions are proposed, by choosing a correct term which is listed. It is possible to index the documents by using terms that are not part of the semantic structure used (iii). If we take the example of a photo of *la Place Royale à Pau*, the expert can choose the term *Place Royale (Pau)*, a specific term which the committee in charge of updating RAMEAU did not choose to include in the thesaurus. Concerning errors linked to homonymy (iv), we limit correction assistance by suggesting, a list of terms containing the term, which reports the error.

5 Conclusion

In this article, we have presented our research work on the structuring and adaptation of a controlled vocabulary as an indexing tool. We worked on a real collection in which we extracted a sub-set of 750 documents and associated descriptive notices. We validated our first experiments with experts of the domain, i.e. librarians. The defined thesaurus representing this indexation work offers a first step for expert users to navigate in the collection through their own representation. Our structure, including a verification phase of the used terms allows in a second step to offer tools for correcting any errors, thereby facilitating the validation of descriptive notices. Our current work focuses on an approach that aims to define a domain ontology based on a thesaurus. We hope to integrate new knowledge characterizing the territory (adding "localized named entities" and links between concepts). The aim is to offer an interface which allows all kinds of user who want to discover a territory described by documents, to navigate through the collection thanks to the domain ontology.

6 References

[1] Schatz B. R., Johnson E. H., Cochrane P. A. and Chen H. Interactive term suggestion for users of digital libraries: Using subject thesauri and co-occurrence lists for information retrieval. In Proceedings of the 1st ACM Digital Library Conference, pp 126–133, Bethesda, US, 1996

[2] Elghoul M., Méthodologie de conception d'un SIAD pour la gestion documentaire, aide à l'indexation, aide à la construction du thésaurus, aide à la recherché et aide à l'apprentissage. PhD – Paris : Université de Paris IX, 1990.

[3] Card S.K., Mackinlay J.D. et Shneiderman B., 'Information visualization', Readings in information visualization: using vision to think, Morgan Kaufmann Publishers Inc., pp. 1-34., 1999

[4] Kohonen, T., Self-Organizing Maps. Berlin, Heidelberg, Springer, 2006

[5] Pepper S., Moore G., "XML Topic Maps (XTM) 1.0 Specification", TopicMaps.Org, Aug. 2001. Available at http://www.topicmaps.org/XTM/1.0.

[6] Ziti M., Baudoin-Lafon M., Hypermedia Exploration with Interactive Dynamic Maps, International Journal of Human Computers Studies, 43: pp 441-464, 1995.

Semantic Annotation and Linking of Competitive Intelligence Reports for Business Clusters

Tomáš Kliegr[1], Jan Nemrava[1], Martin Ralbovský[1], Jan Rauch[1], Vojtěch Svátek[1], Marek Nekvasil[1], Jiří Šplíchal[2], Tomáš Vejlupek[2]

[1] University of Economics, Prague, Dept. Information and Knowledge Engineering, Winston Churchill Sq. 4, 130 67 Praha 3, Prague, Czech Republic
[2] Tovek spol. s r.o., Chrudimská 1418/2, Praha 3, Prague, Czech Republic
tomas.kliegr@vse.cz, nemrava@vse.cz, ralbovsm@vse.cz, rauch@vse.cz, svatek@vse.cz, nekvasim@vse.cz, splichal@tovek.cz, vejlupek@tovek.cz

1 Introduction

Competitive intelligence (CI) is an ethical business discipline that supports decision makers in understanding the competitive environment. Its main vehicle are *CI reports*, which are prepared on the basis of open sources such as web pages, articles or business registries. A *business cluster* is a geographic concentration of interconnected businesses, suppliers, and associated institutions in a particular field. CI efforts within clusters are more complicated than those within individual companies, as different cluster members may perceive the market situation differently. On the other hand, the cost of CI can be shared across the cluster, which is particularly important for SMEs.

Enriching CI reports with *semantic structures* is a natural way to support easier *retrieval* of relevant textual information by the decision makers (among other, via accomodating to their existing mindsets) and for creating *business maps*. As CI reports are knowledge-rich but condensed documents, their 'semantization' is feasible through *authoring-based manual annotation*, though assistance by automated procedures is desirable. As the most important (not necessarily linear) steps towards a semantic repository for CI reports on a given domain and/or business cluster we consider: (1) ontology design; (2) ontology population; (3) ontology-based text annotation; (4) interlinking.

Within the joint effort of Tovek and the University of Economics, Prague (UEP), in the course of one school year (2007-8), undergraduate students were trained to collect and assemble information relevant for CI goals as well as to master several knowledge technology tools. A base of over 70 annotated CI reports arose by the coordinated effort of student teams; nearly 300 students got involved overall in the (joint) role of report writers, annotators and 'ontologists'. The average size of a textual report was about 3000 words; there were, on average, several tens of annotations per report, each typically spanning over one or few sentences or paragraphs. Three domains, in which business clusters explicitly exist or can potentially be formed, were addressed: *packaging industry*, *glass industry* and *information industry*. Every cluster was examined from the point of view of about 20 key organisations. For each domain a specific domain ontology was built, taking a *core CI ontology* as start-up. The underlying *CI model* for all three domain-specific studies was that of *Porter's Five Forces*, which is a business

methodology for qualitative evaluation of company's strategic position [1]. In accordance with this model, the reports primarily focused on the following issues: the threat of *new entrants*, the bargaining power of *consumers*, the threat of new *substitute products*, the bargaining power of *suppliers* and the rivalry of *existing competitors*.

2 Semantic CI Report Workflow

The process of semantic CI report creation is depicted in Fig. 1: boxes correspond to activities, solid arrows to interdependencies involving direct data/artifact flow among activities and dashed arrows to interdependencies without direct data/artifact flow. The activities on the left-hand side (with underlined text) were carried out by CI experts from Tovek; the 'merging' activity in the middle bottom (with slanted text) was carried out by experienced knowledge engineers (and teachers) from UEP; all the remaining activities were carried out by UEP students under modest supervision of teachers. Two 'semantic' *software tools* were used: Ontopoly and Tovek Topic Mapper (TTM).

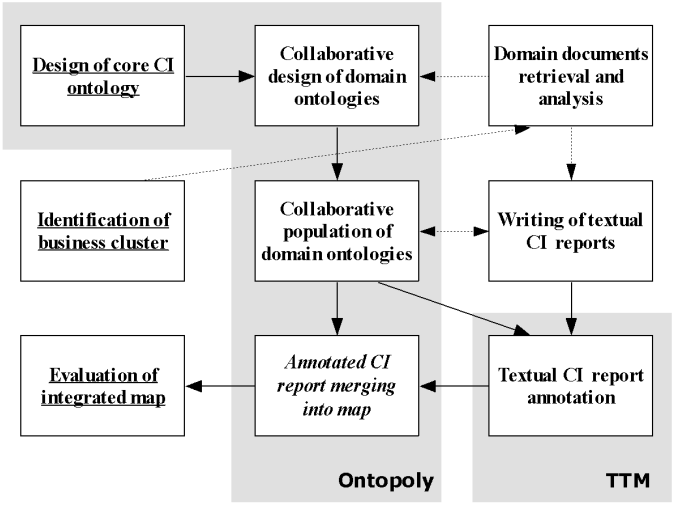


Fig. 1. Schema of workflow

The initial impetus was from the CI experts who designed a kind of *core ontology of CI* (covering, in particular, numerous notions defined in Porter's Five Forces) and also suggested interesting *business clusters*. The student teams bid for companies from the given domain pool and then started to collect *textual documents* such as news articles and web pages that were relevant with respect to 'their' company. Information collected from these resources was the basis for *writing textual CI reports*. At the same time, the students collaboratively extended the core CI ontology with *domain-specific concepts and relations* and then *populated* it with *instances* such as companies, products or people and their interrelationships. The textual reports were then loaded into the

TTM tool and *annotated* with ontology entities. A selected (by quality) subset of annotated reports was then *merged*, together with the underlying ontology, into a larger *CI map* allowing to access the full documents, which was submitted back to the CI experts. The final phase, *evaluation* in the business context, is ongoing.

Two semantic technology tools were exploited: Ontopoly and Tovek Topic Mapper. Both tools use the lightweight formalism of *Topic Maps*.[3] *Ontopoly*[4] is a generic tool for editing and browsing Topic Maps ontologies; for collaborative ontology design and population it had however to be adapted so that students could remotely update ontology data stored on a PostgreSQL server. *Tovek Topic Mapper* is a freely-downloadable[5] tool for ontology-based text annotation developed by Tovek. For the students, the typical amount of work on the application using the semantic tools (i.e. not counting training and textual CI report writing) was about 10 hours per person.

We collected several important observations from the process. A positive one was that the collaborative ontology editing phase was that the number of duplicities thus introduced is quite low. However, the ontology structure, especially the naming policy, was a bit messy at places. The annotation phase sometimes produced results (annotated chunks of text) of very uneven granularity.

3 Conclusions and Future Work

The presented project is probably one of the first attempts to systematically and massively apply semantic technologies in connection with textual CI report authoring, especially in the context of large business clusters. As side effect, it may also serve as generic testbed for collaborative ontology design; this nowadays popular approach[6] has probably not been extensively tested in connection with the Topic Maps formalism yet.

An inherent problem is the reserved attitude of some members of business clusters to joint CI undertakings in general and to the use of semantic technologies for this purpose in particular, which makes the industrial feedback lengthy. On the other hand, there is ample room for improving the quality of results, which would presumably lead to lowering the barriers between the academic project and the business clusters. Several updates will be effectuated in the next round: the quality of ontology design and population should rise thanks to more substantial training of students in *ontological engineering*; the form of annotations will be more uniform thanks to the availability of *annotation guidelines* (dealing with granularity issues etc.); a *content management system* will help manage documents more easily; finally, a *named entity recognition tool* will assist the students, allowing to create annotations more rapidly.

The research was partially supported by the the CSF project no. 201/08/0802.

References

1. Porter, M.E.: Competitive Advantage, The Free Press, New York (1985).

[3] `http://www.topicmaps.org/`
[4] `http://www.ontopia.net/solutions/ontopoly.html`
[5] From `http://www.tovek.cz/produkty/topicmapper.html`
[6] Cf. `http://km.aifb.uni-karlsruhe.de/ws/ckc2007/challenge.html`

Distinguishing general concepts from individuals: An automatic coarse-grained classifier

Davide Picca

University of Lausanne–CH-1015 Lausanne -Switzerland
davide.picca@unil.ch

Abstract. Named entity recognizers are unable to distinguish if a term is a general concept as "scientist" or an individual as "Einstein". In this paper we explore the possibility to reach this goal combining two basic approaches: (i) Super Sense Tagging (SST) and (ii) YAGO. Thanks to these two powerful tools we could automatically create a corpus set in order to train the SuperSense Tagger. The general F1 is over 76% and the model is publicly available.

1 Introduction

In the ontology field, structured information often relies on a structured taxonomy. We assume that, from the ontology engineering perspective, instances of concepts are the leaves of taxonomic structures, since they cannot be further sub-categorized. Named Entity Recognition (NER) can be an useful step for broad-coverage ontology engineering. For example, named entity categories could be used for ontology population and organization. Nevertheless, classical NER systems do not distinguish *universals categories* from *particulars* other than named entities and often the tagset is limited to few categories. It is a main limitation for ontological applications.We present in this paper an automatic classifier for obtaining a coarse-grained distiction between concepts and instances providing more categories.

2 Experiment and results

For our experiment, we used Semcor [2]. The Semcor corpus is a subset of the Brown corpus tagged with WordNet senses, and consists of more than 670,000 words from 352 text files. WordNet defines 45 lexicographer's categories, also called *supersenses* [1], used by lexicographers to provide an initial broad classification for the lexicon entries. SuperSenseTagging is the problem to identify terms in texts, assigning a "supersense" category (e.g. `person`, `act`) to their senses in context and apply it to recognize concepts and instances in large scale textual collections of texts. Sense tagging was done for nouns, verbs, adjectives and adverbs categories. We tested each element of the entire WordNet Supersense tagset in order to find the most adequate to be subdivided in concept and instance. We started from the assumption that only concrete categories can

have a clear distinction between general concept and instance concepts. In our investigation we found six concrete categories and they are the three traditional ones (person, group, location) plus three others (animal, food, artifact).

Successively, we subdivide each category into two sub-categories so that now a term like "president" is tagged as *noun.person_Concept* and a term like "Bill Clinton" as *noun.person_Instance*. In order to automatize this task for all categories, we adopted the following strategy. For each term belonging to the concrete categories, we check if it appears in the YAGO entity dataset [3] otherwise if the term is not found in YAGO, it has to satisfy these following conditions to be tagged as *instance*:

- The part of speech belongs to a noun category as "NN", "NNS", "NNP" or "NNPS".
- The first letter of the term is capital.
- The term does not come before a full stop.

Upon a total of 6407 just almost $\frac{1}{3}$ have been found in YAGO for a total of 2062 terms found as depicted in Table 1. YAGO knows over 1.7 million entities (like persons, organizations, cities, etc.). YAGO exploits Wikipedia's info-boxes and category pages.

	total number of
concepts	11298
instances	6407
instances found in YAGO	2062
instances found using heuristic	4345

Table 1. Total number of concepts and instances

The final result is a tagged sentence as follows:

Bill NNP B-noun.person_Instance Clinton NNP I-noun.person_Instance has 0 0 been 0 0 a 0 0 president NN B-noun.person_Concept of 0 0 USA NNP B-noun.location_Instance.

Following this method, we got very impressive results. We randomly took a sample of the corpus ($\frac{1}{3}$ of the total) and we manually verified the correctness of results. We found 100% of terms correctly classified.

Then, we trained the SST engine with the corpus generated so far, and we optimized the required parameters by adopting a cross validation technique. As for the English settings developed by [1], the best results have been obtained by setting 50 trials and 10 epochs to train the perceptron algorithm.

3 Evaluation

We evaluated the performances of the SST generated so far by adopting a n-fold cross validation strategy on the Semcor adopted for training. Results for chosen

categories are illustrated in Table 2, reporting precision, recall and F1 for any Supersense. If we cast a deeper glance at the Table 2, we can clearly notice that for some category the F1 is exceptionally high. Some of those best categorized categories are really essential for ontology engineering. For example, important labels as *noun.person*, *noun.group* or *noun.location* achieve results higher than 70%. We obtained a general F1 of 0.76%. For some categories we got a F1 over 0.80% as *noun.person_Instance* (F1 0.90%) or *noun.person_Concept* (F1 0.81%)

Category	Recall	Precision	F1
noun.animal_Concept	0.712737	0.76685	0.738763
noun.animal_Instance	0.416667	0.793651	0.545809
noun.artifact_Concept	0.726305	0.737578	**0.731895**
noun.artifact_Instance	0.596154	0.646576	0.620021
noun.food_Concept	0.687179	0.720468	0.7034
noun.food_Instance	0.444444	0.5	0.457143
noun.group_Concept	0.729078	0.731686	**0.730376**
noun.group_Instance	0.683712	0.703622	0.693496
noun.location_Concept	0.682471	0.653416	0.6676
noun.location_Instance	0.752593	0.800006	**0.775557**
noun.person_Concept	0.8384	0.804964	0.821325
noun.person_Instance	0.927861	0.881445	**0.904052**

Table 2. Recall, precision and F1 for each category

4 Conclusion and future work

In this paper we presented a new Supersense Tagger able to distinguish named entities from concepts in texts achieving reasonably high accuracy.

These results are encouraging and this research deserves further investigations. First of all we are going to develop automatic techniques based on parallel corpora to develop SST for other languages, such as German and French, without exploiting any labeled data.

References

1. M. Ciaramita and M. Johnson. Supersense tagging of unknown nouns in wordnet. In *Proceedings of EMNLP-03*, pages 168–175, Sapporo, Japan, 2003.
2. G Miller, C Leacock, R Tengi, and R Bunker. A semantic concordance. *Proceedings of the 3rd DARPA Workshop on Human Language Technology Workshop*, Dec 1993.
3. Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 697–706, New York, NY, USA, 2007. ACM Press.

Pattern-Based Representation and Propagation of Provenance Metadata in Ontologies

Miroslav Vacura and Vojtěch Svátek

University of Economics, W. Churchill Sq.4, 130 67 Prague 3, Czech Republic
{vacuram|svatek}@vse.cz

1 Motivation

Future semantic web applications will rely on multiple ontologies and data collections discovered and assembled at run time into the target knowledge structures. Prototype tools for semantic data retrieval, selection, evaluation and integration already exist, such as Watson.[1] In such dynamic settings, designers of end-user applications however have limited control over the origins of information that influences the results of retrieval and inference. *Provenance metadata* associated with A-box axioms (facts) as well as T-box/R-bow axioms will thus add significant value to such results. As an example from the domain of semantic multimedia,[2] let us consider a user searching for shots of successful actions of England's football player Steven Gerrard. The web-scale search and reasoning process, triggered by query concepts such as 'Gerrard', 'football player', 'action' and 'success', may stray to the pool of semantic information about the Australian rugby (football) player Mark Gerrard. The returned shots thus may feature, in addition to S. Gerrard's goals, also M. Gerrard's *tries*.[3] Such shots may obviously look strange to the user; even if they are finely labelled as tries, a user, ignorant of rugby rules, may be puzzled why a 'mere try' is considered as successful action. Getting a deeper understanding of the problem via detailed inference tracking may be tedious without provenance information on axioms. On the other hand, knowing that the ultimately derived RDF fact (A-box axiom) "Shot123 depicts_situation Sit123" (the resource Sit123 being further connected with the resource *MarkGerrard* and concept *SuccessfulAction*) is declared as having "RugbyOntology" as part of its provenance information (inherited among other from the T-box axiom "*Try* subclassOf *SuccessfulAction*"), the user may simply choose to filter the results using this provenance information.

In the extended abstract we suggest an ontology pattern for representing provenance metadata relying on the forthcoming OWL 2 specification,[4] and propose a simple mechanism for propagation of such metadata. More thorough discussion and a detailed example can be found in a working draft [2].

[1] http://watson.kmi.open.ac.uk
[2] Our prime domain of interest in the *K-Space* project, http://www.kspace.eu.
[3] A try is the major way of scoring points in rugby football.
[4] Being developed by the OWL WG, see http://www.w3.org/2007/OWL/.

2 Pattern for Representation of Provenance in Ontologies

In order to assign provenance information to axioms of a 'base' ontology we need to *reference* them. For this [3] discussed three options: 1) to reify all axioms in the base ontology; 2) to include metadata in annotation properties of the base ontology; 3) to annotate all axioms in the base ontology with an URI and to refer to this identifier in a meta-level ontology. The first approach exposes axioms of the base ontology as individuals of the meta-level ontology but can be computationally difficult. The second approach is relatively simple but it treats provenance outside the logical semantics of OWL; axioms have to be annotated indirectly through entities. The major drawback of the third approach was assumed to be the necessity to extend the OWL 1.0 standard. Nowadays we can however rely on the forthcoming *OWL 2* standard, in which it will be possible to assign URIs to axioms.

The ontology pattern is depicted in simple form in Fig. 1 (a more detailed diagrams with examples is in [2]). The reification level of the provenance ontology consists of class `OntologyAxiom` with subclasses `RBoxAxiom`, `TBoxAxiom`, and `ABoxAxiom`. Instances of these classes are reifications of axioms of the base ontology. The (functional and injective) reification relation R_p is defined as follows: let a be an individual of the provenance ontology and let α be an axiom of the base ontology; then $R_p(a, \alpha)$ iff α is an axiom annotated with a unique identifier URI_α, a is an instance of class `OntologyAxiom` and its data property `AxiomURI` has value URI_α. Reified axioms are then assigned provenance information by linking to individuals of class `ProvenanceAtom` using the relation `prov-for`. As this relation is N:N, a reification of an axiom can be assigned *multiple provenance information atoms* (if the same axiom was included in multiple original ontologies) and of course *multiple axioms* can be assigned a single provenance information atom. Individuals of class `ProvenanceAtom` have provenance information as data properties, e.g. *dc:creator* (omitted in the diagram). Each provenance atom individual is in relation `prov-type` with individuals of class `ProvenanceType`, used to define what provenance model we adopt (for example, Dublin Core). For provenance types we can define the list of attributes that each standard supports using the class `ProvenanceAttribute` linked to class `ProvenanceType` by relation `prov-attr`. This approach allows us to use annotations defined by *multiple provenance models* in a single ontology.

3 Propagation of Provenance Metadata in Ontologies

Not only asserted but also *inferred* axioms obviously need provenance information. Let α be an axiom inferred from the (presumably, merged) ontology O, originally with no provenance information. We follow up with [1], which introduces the *justification* for inferred axiom α in ontology O, $\mathsf{JUST}(\alpha, O) \subseteq O$, as such fragment of O that $\mathsf{JUST}(\alpha, O) \models \alpha$ and $\forall O'((O' \subset \mathsf{JUST}(\alpha, O)) \rightarrow (O' \not\models \alpha))$. Let $O_{AJ(\alpha)}$ be the *union of all justifications* of α in O.

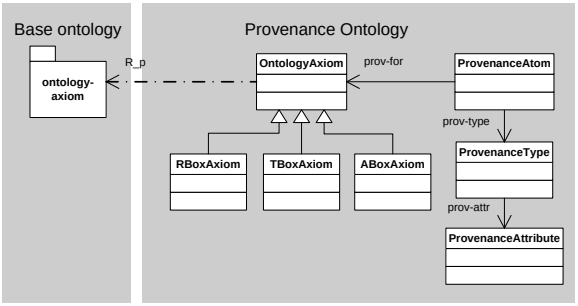


Fig. 1. Provenance representation pattern

We first annotate α with a new unique URI, URI_α. A new individual a of class `OntologyAxiom` is then introduced to the provenance ontology as reification of α, with the data property `AxiomURI` filled with URI_α. Now we consider the set of axioms $O_{AJ(\alpha)}$ and get its respective set of reifications at the first level of the provenance ontology, as $R_p^{-1}(O_{AJ(\alpha)})$. These reifications have provenance information assigned through `prov-for`; we denote the respective set of provenance atoms as $\mathsf{prov\text{-}for}^{-1}(R_p^{-1}(O_{AJ(\alpha)}))$; it pertains to all axioms from the justifications for our inferred axiom α. Since a is the reification of α, for every individual x from $\mathsf{prov\text{-}for}^{-1}(R_p^{-1}(O_{AJ(\alpha)}))$ we add to the provenance ontology the instance of relation $\mathsf{prov\text{-}for}(x, a)$. The whole process is in Fig. 2.

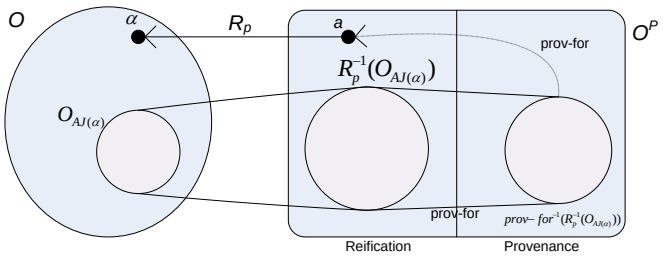


Fig. 2. Provenance propagation

The research is supported by the EC under the project K-Space (FP6-027026).

References

1. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding All Justifications of OWL DL Entailments. In: Proc. ISWC/ASWC, 267–280. Springer, 2007.
2. Vacura, M., Svátek, V.: Representation and Propagation of Provenance Metadata in Ontologies. Working draft, online at `http://nb.vse.cz/~vacuram/doc/pp.pdf`.
3. Vrandecic, D., Völker, J., Haase, P., Tran, D.T., Cimiano, P.: A Metamodel for Annotations of Ontology Elements in OWL DL. In: Proc. 2nd Workshop on Ontologies and Meta-Modeling, Karlsruhe, 2006.

COGNITIVE REENGINEERING OF EXPERT'S KNOWLEDGE BY THE IMPLICIT SEMANTICS ELICITATION

Alexander Voinov [1], Tatiana Gavrilova [2]

[1]Saint-Petersburg State Polytechnic University,
Politechnicheskaya, 29, St.Petersburg, Russia, avoinov@gmail.com
[2]Graduate School of Management, Saint-Petersburg State University,
Volhovsky per., 3, St.Petersburg, Russia, tgavrilova@gsom.pu.ru

Recent years have witnessed increased interest to knowledge modeling in cognitive architecture. Increasingly, the focus has been put on exploring the structures of individual knowledge spaces, using such models as ontologies, frames, rules and semantic networks [1], [2].

Every specific subject domain includes its own combination of the formalized and reproducible knowledge on the one hand and the unique professional experience of its experts on the other hand. The more is the role of the latter in a particular domain the more important is taking into consideration the expert's system of meanings.

Subjective scaling is a formalized interview, where a respondent is given a series of questions with the closed form lists of answers. The list of answers is the same for all the questions. Each question contains an invariant part – the instruction – and a varying pair of stimuli which are subject to comparison. There are certain methodological constraints on the choice of stimuli, e.g. all of them should be taxonomically homogeneous. But in general this choice is more the result of art than of a formal procedure. All pairs of stimuli are presented to the respondent in a specially arranged random order, so that each stimulus appears in the sequence of interview with a similar frequency.

In this paper we illustrate both the classical and the metaphoric versions of subjective scaling by studying the 'domain' of programming languages.

The basic concept space, i.e. the set of elements, representing the domain of interest, was built of a list of the most popular and well-known programming languages.

For the reasons of the limitation of the poster space, we present here the complete result and interpretation of only one of our experiments.

Our respondent is a high level system programmer, working in a team, which develops software tools for artificial intelligence. His professional programming language is C (not C++). Processing of his answers for the classic subject scaling experiment has resulted in the following graph of the two most dominant axes (see Fig.1).

This graph evidently reflects the usual, generally accepted distribution of programming languages into classes by their intent and performance. This classification reveals little new with respect to the shallow, verbal knowledge of everybody, who is aware of this domain.

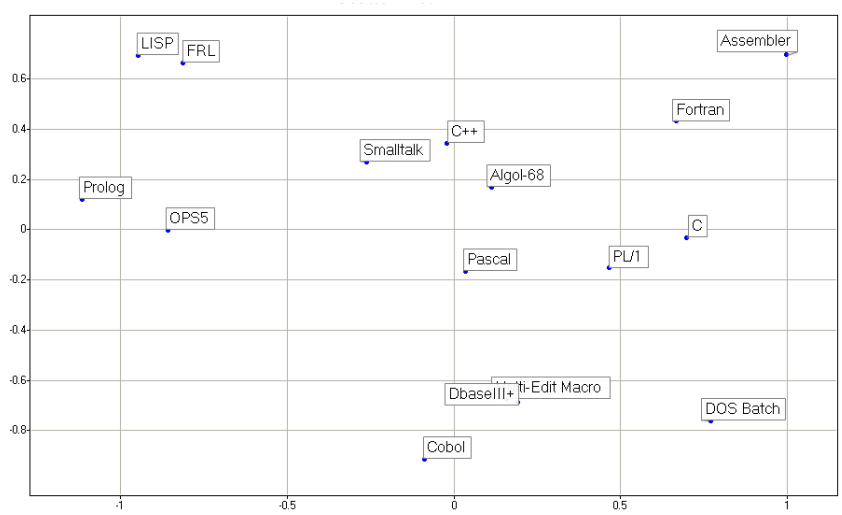


Fig. 1. Results of Classical Subjective Scaling

The axes on the Fig.1 can be interpreted as ‘declarative’ vs ‘imperative’ languages (horizontal) and ‘performance’ vs ‘slow’ (vertical).

For the metaphoric version of the experiment we used three versions of ‘metaphoric spaces’: animals, cars and tale/folklore heroes.

The interview procedure was adjusted to include only the comparisons of the languages with the different metaphors, like:

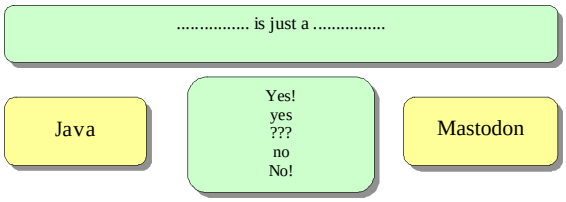


Fig. 2. The Metaphoric Subjective Scaling

The categories were also numbered to give maximum similarity to “Yes!” and maximum dissimilarity to “No!”. The sets of respondents’ answers were processed with correspondingly modified methods of multidimensional scaling.

The most interesting results were obtained using the metaphoric world of the fairy tales heroes. In our paper we present just one of the collected results. The set of chosen “heroes” of tales, animation movies and the children’s literature was relevant and familiar to the Russian-speaking respondents.

The graph, visualizing the answers of one of our respondents (same as at the Fig.1), is shown at the Fig. 3.

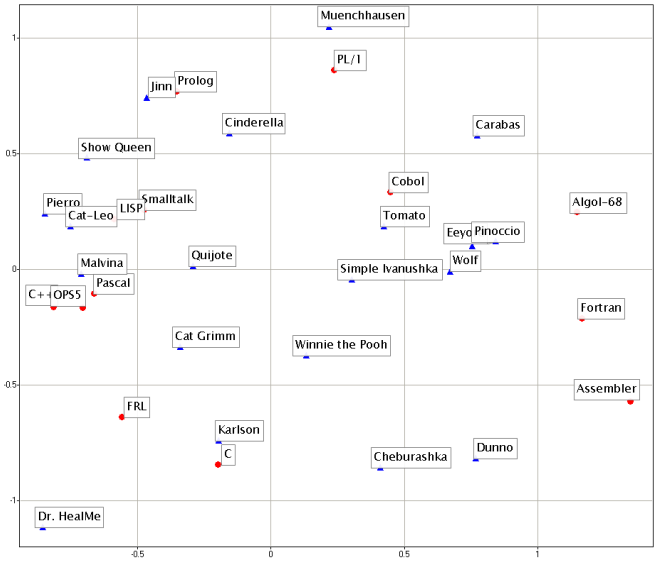


Fig. 3. The Results of the Metaphor-Driven Subjective Scaling

We see an evident difference in the representations because the metaphorical result elicits the more personal implicit semantics and individual respondent's attitude to the stimuli, that reflects his professional skills and expertise.

The meaning of the two major axes was verbalized as “Crude” vs “Refined” creatures and “Extraversion” vs “Introversion”. It is rather strange features of the attitude to the languages but they exist.

The strongest point of the described approach is the ability to elicit the hidden cognitive constructs that reengineer the whole semantic space of expert’s knowledge patterns. These hidden constructs create the real conceptual model of expert’s vision. Metaphorical scaling reveals that implicit priorities, values and attitudes.

Such methods may be not as often used as the other knowledge engineering (KE) techniques. It is rather time-consuming and exotic as needs the choosing of the proper metaphor and finding of the relevant set of key concepts or stimuli. But as a complimentary method it may facilitate the general KE strategy with a novel bias.

The value of this technique to ontology engineering concludes in an ability to reveal unexpected classifications of concepts, which are inherent to practicing experts in that domain. The ontological classes or categories may be based on the different common features acquired by the metaphorical approach.

References

1. Adeli, H. Knowledge Engineering. McGraw-Hill, New-York (1994)
2. Gómez-Pérez, A., Fernández-López, M., Corcho, O. Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web, Springer (2004)

EKAW 2008
16th International Conference on
Knowledge Engineering and Knowledge Management
Knowledge Patterns

29th September-3rd October 2008

Acitrezza, Catania, Italy