The 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference

Workshop 10

International Workshop on Emergent Semantics and Ontology Evolution

Workshop Organizers:

Luke Liming Chen, Philippe Cudré-Mauroux
Peter Haase, Andreas Hotho, Ernie Ong

12 Nov. 2007
BEXCO, Busan KOREA

ISWC 2007 Sponsor

Emerald Sponsor

Gold Sponsor

Silver Sponsor

We would like to express our special thanks to all sponsors

ISWC 2007 Organizing Committee

General Chairs

Riichiro Mizoguchi (Osaka University, Japan)
Guus Schreiber (Free University Amsterdam, Netherlands)

Local Chair

Sung-Kook Han (Wonkwang University, Korea)

Program Chairs

Karl Aberer (EPFL, Switzerland)
Key-Sun Choi (Korea Advanced Institute of Science and Technology)
Natasha Noy (Stanford University, USA)

Workshop Chairs

Harith Alani (University of Southampton, United Kingdom)
Geert-Jan Houben (Vrije Universiteit Brussel, Belgium)

Tutorial Chairs

John Domingue (Knowledge Media Institute, The Open University)
David Martin (SRI, USA)

Semantic Web in Use Chairs

Dean Allemang (TopQuadrant, USA)
Kyung-Il Lee (Saltlux Inc., Korea)
Lyndon Nixon (Free University Berlin, Germany)

Semantic Web Challenge Chairs

Jennifer Golbeck (University of Maryland, USA)
Peter Mika (Yahoo! Research Barcelona, Spain)

Poster & Demos Chairs

Young-Tack, Park (Sonngsil University, Korea)
Mike Dean (BBN, USA)

Doctoral Consortium Chair

Diana Maynard (University of Sheffield, United Kingdom)

Sponsor Chairs

Young-Sik Jeong (Wonkwang University, Korea)
York Sure (University of Karlsruhe, German)

Exhibition Chairs

Myung-Hwan Koo (Korea Telecom, Korea)
Noboru Shimizu (Keio Research Institute, Japan)

Publicity Chair: Masahiro Hori (Kansai University, Japan)

Proceedings Chair: Philippe Cudré-Mauroux (EPFL, Switzerland)

Metadata Chairs

Tom Heath (KMi, OpenUniversity, UK)
Knud Möller (DERI, National University of Ireland, Galway)

Preface

The Semantic Web and collaborative tagging are two complementary approaches aiming at making information search, retrieval, navigation and knowledge discovery easier. While the Semantic Web enforces semantics top-down via the use of ontologies, collaborative tagging tries to obtain semantics in a bottom-up fashion. Del.icio.us and flickr are success stories of collaborative tagging; the winners of the Semantic Web Challenge demonstrate the success of the Semantic Web. Still, both approaches face open issues. For the Semantic Web, ontology engineering, in particular, large-scale ontology construction, has been a bottleneck. While effort and progress have been made in ontology matching, alignment, versioning and learning, it has become clear that constructing large ontologies requires collaboration among multiple individuals or groups with expertise in specific areas. Also critical is the ontology evolution in the open, dynamic Web environment in order to keep pace with the Web dynamics. For collaborative tagging, tags (metadata) can be generated in large-scale and capture users collective wisdom. However, large-scale tagging usually degrades the performance of re-findability due to the ambiguity of uncontrolled vocabulary and the flat structure of tag soup. In such a case tagging alone is not helpful at all for solving the problem. Bundles, classification, relations or tagging of tags are some promising ways to enforce some kinds of structure for tags in order to enable scalability and findability.

We believe that the mashup and synergy of the two paradigms is the key to create large-scale semantic and intelligent content. The vision is that we should and can (1) derive emergent semantics from community-based collaborative interaction as demonstrated by Web 2.0 applications, in particular, folksonomic tagging; (2) extract and formally model emergent semantics in structures, such as ontologies; (3) construct and evolve ontologies as emergent semantics from collaborative applications are of dynamic nature; and (4) enhance collaborative applications with formal ontological structures, and enable large scale semantic Web applications.

Against this background, we organize this workshop, aiming to provide a forum for researchers and practitioners in the relevant fields of the Semantic Web, ontology engineering, folksonomy, social Web, artificial intelligence, machine learning, information integration and relevant application areas to discuss the current state of the art and open research problems in emergent semantics and ontology evolution. This proceeding contains nine research papers reporting the latest research activities and initial results in this interdisciplinary area, which, while some of them are still at the early stage, offer future research directions, inspirations and visions.

We expect that, through the workshop, understanding on the emergent semantics and ontology evolution be deepened, collaborations between researchers and/or teams be formed, and more attention and effort be drawn to this emerg-

ing research area. Last but not least we thank the PC members and additional reviewers for their useful comments on the submitted papers, all authors for inspiring papers, the audience for the interest in this workshop, the local organizers from the ISWC 2007, and the Workshop Chair.

November 2007

Luke Liming Chen
Philippe Cudré-Mauroux
Peter Haase
Andreas Hotho
Ernie Ong

Organization

Organizing Committee

Luke Liming Chen, University of Ulster, UK
Philippe Cudré-Mauroux, EPFL – Lausanne, Switzerland
Peter Haase, Institute AIFB, University of Karlsruhe, Germany
Andreas Hotho, KDE Group, University of Kassel, Germany
Ernie Ong, SAP CEC Research Centre, Belfast, UK

Program Committee

Andreas Abecker, FZI, Germany
Karl Aberer, Swiss Federal Institute of Technology (EPFL), Switzerland
Harith Alani, University of Southampton, UK
Ciro Cattuto, University of Roma La Sapienza, Italy
Stefan Decker, DERI, Galway, Ireland
Manfred Hauswirth, DERI, Galway, Ireland
Peter Mika, Yahoo, Barcelona, Spain
Natasha Noy, Stanford University, USA
Daniel Oberle, SAP Research Karlsruhe, Germany
Steffen Staab, University of Koblenz, Germany
Ljiljana Stojanovic, FZI Karlsruhe, Germany
Leo Sauermann, DFKI, Germany
Harald Sack, University of Jena, Germany
Marta Sabou, Knowledge Media Institute, UK
Luc Steels, Free University of Brussels (VUB), Belgium
Frank van Harmelen, Vrije Universiteit, The Netherlands
Denny Vrandecic, AIFB, University of Karlsruhe, Germany

Table of Contents

Invited Talks

Research Papers

Emergent Semantics Systems

Karl Aberer

School Of Computer and Communication Sciences
EPFL – Switzerland
karl.aberer@epfl.ch

Abstract

Until recently, most data interoperability techniques involved central components, e.g., global schemas or ontologies, to overcome semantic heterogeneity for enabling transparent access to heterogeneous data sources. Today, however, with the democratization of tools facilitating knowledge elicitation in machine-processable formats, one cannot rely on global, centralized schemas anymore as knowledge creation and consumption are getting more and more dynamic and decentralized. Peer Data Management Systems (PDMS) implementing semantic overlay networks are a good example of this new breed of systems eliminating the central semantic component and replacing it through decentralized processes of local schema alignment and query processing. As a result semantic interoperability becomes an emergent property of the system.

In this talk we provide examples of both structural and dynamic aspects of such emergent semantics systems based on semantic overlay networks. ¿From the structural perspective we can show that the typical properties of self-organizing networks also appear in semantic overlay networks. They form directed, scale-free graphs. We present both analytical models for characterizing those graphs and empirical results providing insight on their quantitative properties. Then we present semantic gossiping, a model for the dynamic reorganization of semantic overlay networks resulting from information propagation through the network and local realignment of semantic relationships. The techniques we apply in that context are based on belief propagation, a distributed probabilistic reasoning technique frequently encountered in self-organizing systems. Finally we will give a quick glance on how this techniques can be implemented at the systems level, based on a peer-to-peer systems approach.

Biographical Sketch

Karl Aberer is a Professor for Distributed Information Systems at EPFL Lausanne, Switzerland, and director of the Swiss National Centre for Mobile Information and Communication Systems (NCCR-MICS). His research interests are on decentralization and self-organization in information systems with applications in peer-to-peer search, overlay networks, trust management and mobile and sensor networks. Before joining EPFL in 2000 he was leading the research division of open adaptive information systems at the Integrated Publication and Information Systems Institute (IPSI) of GMD in Germany, which

he joined in 1992. There his work concentrated on XML data management and cross-organizational workflows. He studied mathematics at ETH Zrich where he also completed his Ph.D. in theoretical computer science in 1991. From 1991 to 1992 he was postdoctoral fellow at the International Computer Science Institute (ICSI) at the University of California, Berkeley. He is member of several journal editorial boards, including VLDB Journal, and conference steering committees. Recently he served as PC co-chair of ICDE 2005, MDM 2006 and ISWC 2007.

Ontology Learning: Where are we? And where are we going?

Paul Buitelaar

DFKI GmbH
Language Technology Lab & Competence Center Semantic Web
Saarbrcken, Germany

Ontology learning concerns the development of automatic methods for the extraction of a domain model from a relevant, i.e. domain-specific data set. In the context of ontology evolution, a specific domain model is already given and the task of ontology learning reduces to the extension or adaptation of this domain model on the basis of a changing underlying data set.

Ontology learning largely builds on methods previously developed in knowledge acquisi- tion, natural language processing and machine learning although with the specific purpose of automatically deriving an ontology, i.e. an explicit, shared and formally defined logical model. Unfortunately, the current state-of-the-art in ontology learning cannot be said to have reached this goal yet, although progress is made on various levels over the last couple of years.

Ontology learning is in fact not really one task but rather a collection of tightly connected subtasks that can be organized in a layered representation of increasing complexity, i.e. term extraction, synonym and translation detection, concept formation, instantiation, relation extraction, paraphrase and rule derivation, axiomatization. On each of these levels, methods and tools have been developed that address one or more subtasks. Methodologies are still needed however that address all subtasks in a coherent way and provide benchmarks for evaluation of methods on all levels, separately and in combination.

Ontology learning tools need to perform well on all levels of analysis, but even this is no ultimate guarantee for being actually useful. In addition to performance considerations, ontology learning tools need to be fully integrated into the knowledge engineering life-cycle, working in the background and providing the human domain expert with relevant input for ontology construction or evolution. Usability of ontology learning tools will thus be measured in terms of productivity of the human domain expert.

Ontology learning until recently has been based mostly on knowledge extraction from textual data, although some work has been done on extraction from tables and other structured data. Currently however, more and more semi-structured data becomes available in the form of Wikis and User Tags that shows a number of advantages for ontology learning as these data sets carry a lot of implicit knowledge (i.e. relations by linking or by social grouping) that can be more easily extracted than similarly implicit knowledge available in textual data. Additionally, more and more ontologies become publicly available that may be used as input by ontology learning tools, possibly in combination with knowledge derived from Wikis and User Tags and from more traditional textual data sets.

Ontology learning is a relatively new field of research, although building on long-standing methods in AI. In the developing context of the Semantic Web it is and will remain a central field of attention as ontologies form the semantic backbone of the Semantic Web, whereas their construction is complex and therefore knowledge- and cost-intensive. Automating this process through ontology learning thus remains an attractive proposition.

The Ontology Maturing Approach for Collaborative and Work Integrated Ontology Development: Evaluation Results and Future Directions

Simone Braun, Andreas Schmidt, Andreas Walter, Valentin Zacharias

FZI Research Center for Information Technologies
Information Process Engineering
Haid-und-Neu-Straße 10-14, 76131 Karlsruhe, Germany
{Simone.Braun|Andreas.Schmidt|Andreas.Walter|Valentin.Zacharias}@fzi.de

Abstract. Ontology maturing as a conceptual process model is based on the assumption that ontology engineering is a continuous collaborative and informal learning process and always embedded in tasks that make use of the ontology to be developed. For supporting ontology maturing, we need lightweight and easy-to-use tools integrating usage and construction processes of ontologies. Within two applications – ImageNotion for semantic annotation of images and SOBOLEO for semantically enriched social bookmarking – we have shown that such ontology maturing support is feasible with the help of Web 2.0 technologies. In this paper, we want to present the conclusions from two evaluation sessions with end users and summarize requirements for further development.

1 Introduction

The first wave of semantic (web) applications has shown that ontologies are well-suited for sophisticated ways of retrieval of relevant resources, but traditional ontology engineering methodologies and tools suffer from the underlying assumption that a few modelling experts have to create an ontology for many users. In order to keep the ontology in line with the intended usage, cumbersome procedures are introduced that lead to delayed and often error-prone updates to the ontology (cf. [1,2]). On the other hand, folksonomies are agile, user-driven approaches, but it is increasingly perceived that folksonomies have their clear limitations when it comes to enhancing resource retrieval. While this trade-off between degree of formalization and degree of participation is often considered to be inevitable, we propose in our research to have a look at how we can support smooth and continuous transitions between the two worlds.

Starting from the insight that building an ontology is essentially formalizing an understanding of a particular domain, we conceive ontology engineering as a continuous collaborative learning process, which we call ontology maturing [3]. In a first step, we have created a conceptual process model structuring this maturing into four characteristic phases, ranging from emergence of ideas, consolidation in communities via formalization up to axiomatization. Based on this model, we have built two applications that support maturing by embedding extension and refinement of ontologies into actual

usage processes. The first application (ImageNotion) supports semantic retrieval and annotation of images in large-scale image archives, the second application (SOBOLEO) provides a semantic enhancement of social bookmarking.

In this paper, we want to present the results of a formative evaluation of these tools with end users and the conclusions for future developments. In section 2, we first briefly present the ontology maturing process model before we sketch the tools and their functionality in section 3. In section 4, we present the results from the evaluation sessions and the conclusions for future enhancements.

2 Ontology Maturing Process Model

Starting point of our ontology maturing process model were the shortcomings of the usual separation of creation and usage processes [4]. While this might be possible in rather static domains, it is not acceptable for dynamic domains, especially when using ontologies for the annotation and retrieval of resources, where contents change fast and the ontology requires a permanent update to cover the available contents. In real world setup, this leads to frustrating situations (which *is* a major problem for acceptance) when users cannot extend the used ontologies by themselves in a work-integrated way, e.g. when they require them for the semantic annotation of images or web-pages. Instead, they are forced to ask ontology experts for the extension and wait for the update of the underlying ontologies, which – in very dynamic domains – can even last until the ontology element has become obsolete again [5].

2.1 A Collaborative and Work-integrated View on Ontology Development

This led us to rethink ontology engineering as a collaborative and work-integrated activity. In this view, users themselves (within, e.g., communities of practice) can modify the underlying ontology of a semantic application, e.g., add new ontology elements or modify existing ones. This new perspective, motivated by constructivist views on learning (see also [6]), views the quality of an ontology within the context of a semantic application as a balance of three different aspects:

a) **Appropriateness**. An ontology needs to be an appropriate representation of the domain with respect to the purpose of the ontologies required for a semantic application so that it is actually useful. This is only possible when we have a tight coupling and immediate mutual feedback between changes to the ontology and use of its elements, e.g., for search or annotations. That means, we need a quick, simple and work-integrated way to adapt and modify the ontologies.
b) **Social Agreement.** An ontology needs to represent a shared understanding among all stakeholders. Thus, successful ontology construction is a social and collaborative learning process within the communities of its users. The involved individuals deepen by and by their understanding of the real world and of an (appropriate) vocabulary to describe it.
c) **Formality.** The formalization of ontologies is not possible completely from scratch. In particular for emerging ideas and concepts, it is not possible to directly integrate

them into an ontology as they are not clearly defined, yet. That means, the development of an ontology underlies a process of continuous evolution where different levels of formality might co-exist within one ontology. The outcome is an adequate level of formality in the ontology, avoiding both overformalization and the inability to apply semantic algorithms.

2.2 The Phases of the Ontology Maturing Process Model

To operationalize this view, we have developed the ontology maturing process model that structures the ontology engineering process into four phases (see Fig. 1):

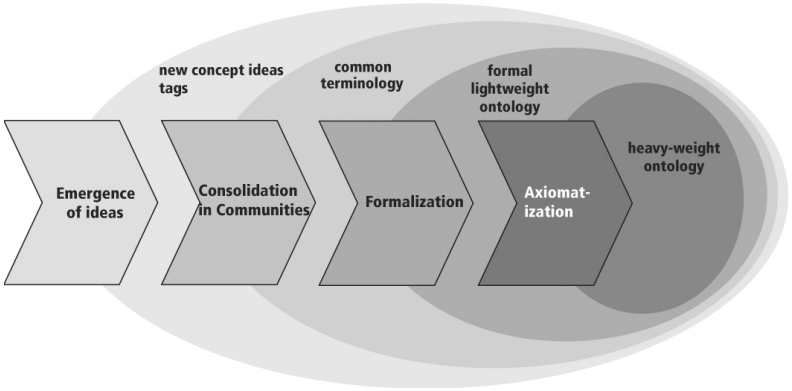


Fig. 1. The four phases of the ontology maturing process model

1. **Emergence of ideas.** New ideas emerge and are introduced by individuals as new concept ideas or informal tags. These are ad-hoc and not well-defined, rather descriptive, e.g. with a text label. They are individually used and informally communicated.
2. **Consolidation in Communities.** Through the collaborative (re-)usage of the concept symbols (tags) within the community, a common vocabulary (or folksonomy) develops. The concept ideas are refined, useless or incorrect ones are rejected. The emerging vocabulary, which is shared among the community members, is still without formal semantics.
3. **Formalization.** Within the third phase, the community begins to organize the concepts into relations. These can be taxonomical (hierarchical) ones as well as arbitrary ad-hoc relations, e.g., in the course of becoming aware of different abstraction levels. This results in lightweight ontologies that rely primarily on inferencing based on subconcept relations.

4. **Axiomatization.** In the last phase the adding of axioms allows and improves for inferencing processes, e.g. in query answering systems. This step requires a high level of competence in logical formalism so that this phase is usually done with the aid of knowledge engineers.

It is important to note that ontology maturing does not assume that ontologies are built from scratch, but can be equally applied to already existent core ontologies used for community seeding. Likewise, this model must not be misunderstood as a strictly linear process; rather real ontology development processes will consist of various iterations between the four different phases.

We identified semantic annotation and retrieval of resources as one possible use case where the ontology maturing process model can demonstrate its potential. We will concentrate on this use case for the rest of this paper, although other semantic applications, e.g. for expert finding or description of web services could benefit from the usage of the ontology maturing process model as well.

3 Tool Support

Our applications realize the ontology maturing process model by offering work-integrated ontology development and an easy-to-use interface to allow the usage of semantic technologies also for "ordinary" people. SOBOLEO allows for the semantic annotation and retrieval of web resources, the ImageNotion tool for the semantic annotation and retrieval of images. In this section, we will give a brief introdution of these applications.

3.1 SOBOLEO

SOBOLEO [7] is the acronym for **So**cial **Bo**okmarking and **L**ightweight **E**ngineering of **O**ntologies. The system's goal is to support people working in a certain domain in the collaborative development of a shared index of relevant web resources (bookmarks) and of a shared ontology that is used to organize the bookmarks. That means, collected bookmarks can be annotated with concepts from the ontology and the ontology can be changed at the same time.

SOBOLEO (see Fig. 2) consists of four major parts: (1) a collaborative real time editor for changing the ontology, (2) a tool for the annotation of web resources, (3) a semantic search engine for the annotated web resources, and (4) an ontology browser for navigating the ontology and the index of the web resources. The users within one community create and maintain one ontology and one shared index of web resources collaboratively.

Thus, the users can create, extend and maintain ontologies according to the SKOS Core Vocabulary [8] in a simple way together with the collection and sharing of relevant bookmarks. If they encounter a web resource, they can add it to the bookmark index and annotate it with concepts from the SKOS ontology for better later retrieval. If a needed concept does not exist in the underlying ontology or is not suitable, the users can modify an existing concept or use arbitrary tags, which are automatically added to the ontology. In this way, new concept ideas are seamlessly gathered when occurring (maturing phase

Fig. 2. The collaborative ontology editor in SOBOLEO

1) and existing ones are refined or corrected (maturing phase 2). The users can structure the concepts with hierarchical relations (broader and narrower) or indicate that they are “related”. These relations are also considered by the semantic search engine. That means, the users can improve the retrieval of their annotated web resources by adding and refining ontology structures (maturing phase 3).

3.2 ImageNotion

ImageNotion [9] is both a methodology based on the idea of the ontology maturing model, and the name of a web-based tool supporting this methodology in the domain of images. An imagenotion (formed from the words image and notion) graphically represents a semantic notion through an image. Each imagenotion may contain additional descriptive information like a label and its synonyms (both possible in different languages), temporal information and links to web pages that contain background information for an imagenotion. Using imagenotions, users do not need to distinguish between concepts and instances in ontologies – a separation of ontology elements often considered artificial. In addition to descriptive information, relations between imagenotion are also possible. Currently we support hierarchical relations (broader and narrower) similar to SKOS [8] – all other relations are called ”unnamed relations” (and correspond to *skos:related*). The aim of the ImageNotion methodology is to guide the process of visually creating an ontology. This ontology will contain imagenotions as semantic elements and relations between them. The main steps of this methodology are based on

the ontology maturing process model. Step 1 is the creation of new imagenotions, step 2 is the consolidation of imagenotions in communities and step 3 is the formalization of imagenotions with rules and relations. Imagenotions from each level of maturity may be used for semantic image annotation. In fig. 3 a user annotates an image showing "Joseph Joffre" (a french general in WWI) with the corresponding imagenotion.

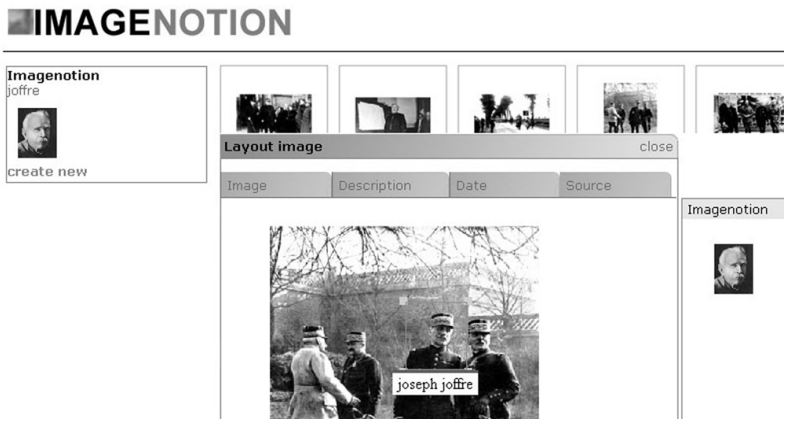


Fig. 3. Semantic annotation of images using imagenotions

One pecularity of communities in the area of semantic image annotation is that we usually have two separate roles and groups of interest: content owners (providing the images) and image users. The content owners use imagenotions for *annotation* to improve the findability. Image users use imagenotions for *searching* images they are interested in, e.g. for commercial usage in media. Both of these groups have to collaborate and thus engage in maturing of imagenotions to improve the quality of semantic annotation of images.

4 Evaluation

As ontology maturing support has to follow a participatory philosophy, it was important to have formative evaluation of our prototypes early on. End-users recently evaluated both tools in two different environments and evaluation settings. In the following, we describe their respective evaluation setups and summarize the results.

4.1 Evaluating SOBOLEO

We evaluated SOBOLEO in two separate sessions. The first evaluation took place from April 16-30, 2007, within the scope of the Collaborative Knowledge Construction Challenge within the Workshop on Social and Collaborative Construction of Structured

Knowledge held at the 16th International World Wide Web Conference[1]. We provided a basic ontology to facilitate getting started and to give thematical orientation for the participants. This ontology was tailored to the research domain as a whole with concepts like 'research topic', 'people', 'institution', 'publication', and 'event'. Everyone was free to participate and contribute information about their research domain. At the end, they were asked to provide feedback. Altogether, 49 users registered and 33 contributed actively to the challenge.

During this evaluation, the participants added in total 202 new concepts and 393 concept relations to the ontology. Further, they collected 155 web resources, which they annotated with 3 concepts per resource on average. None of the users had the opportunity to meet other users using SOBOLEO at the same time. Thus, the chat functionality was barely used; only for testing.

Summarizing the feedbacks, the participants appreciated the ease-to-use of SOBOLEO and having a shared ontology. They emphasized in particular the editor's real-time nature. The users further enjoyed the simple way for annotating web resources with concepts or tags, which are then automatically added. Thus, to have the possibility to integrate not yet well defined concepts but somthing like "*starter concepts*" and, in this way, to "*get the ontology building almost for free*". For improving SOBOLEO, the users pointed out several times that they missed a personal view on the data, i.e. on the own annotated resources but also on the ontology (especially in case of a growing and dispersing user base). Although the users appreciated the messages/chat pane informing about changes and for communication with other users, the users expressed the wish to have more possiblitites to discuss and be informed about modification (on "own" data) by other users. Thus, to gain more translucence and awareness, especially as they could not experience working together simultaneously. A further aspect was to have better support for identifying or suggesting conflicts, synonymous concepts and broader-narrower relations in order to facilitate the maintenance of the ontology.

The second evaluation of SOBOLEO took place within the scope of the project "Im Wissensnetz"[2] (*"In the Knowledge Web"*). This evaluation was especially intended to test usability (especially goal/task support) of SOBOLEO and was assisted by thinking-aloud techniques and screen recording tools. Within two one-hour sessions, four users had to carry out specific tasks simulating the usage of SOBOLEO within their daily work activities. Half of the users were researchers of the rapid prototyping domain and half of them patent experts for German research. All of them were unexperienced in ontology development. We provided a basic ontology with 31 concepts to start with that was thematically tailored to the rapid prototyping domain.

During the second evaluation, the four users created 6 new concepts. This low number can be traced back to the given tasks, which did not demand the explicit creation of new concepts. Instead the tasks were tailored to gain orientation within the ontology by letting the users place or add synonyms to existing concepts. Thus, the users added 11 synonyms and 21 concept relations. During the annotation specific tasks, they collected in total 42 web resources, which they annotated with 2.5 concepts per resource in average.

[1] `http://km.aifb.uni-karlsruhe.de/ws/ckc2007`
[2] `http://www.im-wissensnetz.de`

The users appreciated SOBOLEO for its easy of use. Some of the users had some problems at the beginning due to their very basic knowledge in ontologies and were confused by the concept editing functionality. But a learning effect could be observed shortly. The chat turned out to be an essentiell utility; especially for simultaneous working. For instance, two users had problems in placing concepts in the given ontology because they had only basic knowledge of the rapid prototyping domain. In consequence, they began to ask their colleagues for help via the integrated chat functionality. Nevertheless, the chat appeared to be too simple. For improvement, the users wished to have a better integration of what is discussed and where the changes are done. Further extended functionalities like chat rooms as well as more documentation to understand how and why decisions and modifications are done (also for later use) were required. This evaluation showed as well that translucence and awareness are crucial factors in collaborative ontology development.

4.2 Evaluating ImageNotion

The first stable version of the ImageNotion tool was evaluated in June 2007 by experienced image annotators and librarians having minimal ontology background within the scope of the IMAGINATION project. Our aim was to evaluate whether they are able to collaboratively create ontologies in combination with the semantic annotation of images using the ImageNotion tool. Six people participated at the workshop. The reference set consisted of 854 images from the preselected domains “world war 1” and “European politicians”. One participant had well-founded background knowledge about semantic formalism; two of the participants (user 2 and 3) had many experiences with tag based annotation systems but no experiences with semantic formalisms and applications. The other three participants were familiar with thesauri, but not with the creation of ontologies or with image annotation systems.

ACTIVITY			
Ontology maturing		**Semantic image annotation**	
Number of created imagenotions	46	Number of annotated images	68
Imagenotions with only one work step	10	Imagenotions used for annotation	26
Number of work steps	115	Number of work steps	110

Table 1. Work steps grouped by type of activity

The results of our evaluation were generated in two hours by the partcipants. Comparing the sum of work steps of all users for ontology maturing activities and for annotation activities, table 1 shows that the number of work steps for the work process ontology maturing (115 steps) is higher than the number of worksteps for the semantic image annotation. This shows the need for a work integrated ontology maturing. From the total number of 46 created imagenotions, 26 imagenotions were directly used for the semantic annotation of images, 10 imagenotions were indirectly used through relations to these imagenotions. 10 imagenotions had only one work step each so that they did not pass the phase one ‘Emergence of ideas’ of our ontology maturing model.

Number of work steps	1	2	3	4	5	6
Descriptive	8	1	19	12	27	8
Relations	7	2	9	10	9	3
Total	15	3	28	22	36	11

Table 2. Maturing of imagenotions

Number of users	1	2	3	4	5
Imagenotions	32	11	1	1	1
Percent	70	24	2	2	2
Total	70	24	2	2	2

Table 3. Collaborative usage

Table 2 shows the aggregated number of work steps of the users for the maturing of imagenotions. All users were able to create relations to other imagenotions. In addition, they added a lot of descriptive information to the created imagenotions. During the workshop, the participants could speak together and discuss available imagenotions. We observed that user 1 (who was very familiar with ontology editing) explained the principes of relations to the other participants. Also, we observed that the usage of links to other web pages in imagenotions improved the background knowledge of the users so that they could in turn add further information, e.g. birthday of persons or relations. Table 3 shows the collaborative usage of imagenotions. Already during the two hours of the evaluation, 24 percent of the imagenotions were used by more than one user and thus entered the phase two of our ontology maturing model "consolidation in communities". Again, a main reason for that was the possibility of the participants to talk about the created imagenotions.

The participants of the workshop were all experts about the domains of their images. Even in such a small group of six participants, we observed a specialization for different topics of interests. Two participants mainly annotated images showing airplanes and therefore created relevant imagenotions while the other participants mainly created imagenotions for persons and events to annotate the corresponding images.

5 Lessons Learnt

The evalutations showed that our tools and the underlying ontology maturing process model achieved a high level of acceptance by the participants. During the evaluation sessions and in subsequent discussions, we identified missing and requested features for our tools. These features cover better support for consolidation, the distinction of local and global information and a better support for the creation of groups to spezialice for a specific topic of interest, which shall be described in more detail in the following sub sections.

5.1 Consolidation Support

Based on our ontology maturing process model, the consolidation phase covers combination and refinement of useful ontology elements and the rejection of incorrect or useless ones. Since consolidation is a process of collaborative work, communication between the members of such a community is one of the main functions that help in these processes. In our evaluations, we identified the need for extended communication functionalities, because the participants in the ImageNotion evaluation discussed offline together and in the SOBOLEO evaluation they used the integrated chat functionalites.

However, a simple chat is not enough. Based on discussions with the end users, we identied the following four diffent areas in these consolidation processes that require the extension of our application with specific tools:

Discussion and Agreement In this area, the participants of the group communicate together discuss about available ontology elements and whether they are useful or not. In addition, in case of similar or even duplicate ontology elements, they discuss whether they should be merged or extended. As the SOBOLEO evaluation showed, a simple chat for all is not enough for that because of too many messages concerning different topics. As a solution, we will extend our tools with a threaded chat system that allows for the separation of discussion topics. In addition, we think that a forum application (e.g. JForum[3]) is helpful for asynchronous discussions, i.e. when the members of a community are not always online at the same time.

In our evaluations, we had to handle a relatively small number of participants. In small groups, it is possible to achieve agreements among the members through direct discussions. In case of bigger groups with ten, fifty or even more participants, direct aggreements through discussion is no longer possible. As we plan to allow for bigger groups in our applications, we will extend them with tools that help in voting about open discussions and in rating the quality of given ideas to achieve agreement.

Execution of Changes This area covers tool support complex operations in the consolidation phase. Especially for ontology elements with similar meaning (e.g. because they were created with descriptions in different languages), we see the requirement to integrate tools that help in handling these complex operations. The merging of two ontology elements requires updates of all ressources that have been annotated so far with one of the concerned ontology elements with the newly created one (see e.g. the HCONE approach ([10]). Instead of forcing users to update all these annotations manually, we will offer automated processes for these tasks. Also for the splitting of an ontology element, e.g. in two different subconcepts, we will care for adequate tools.

Dissemination and Creating Awareness Tools for the dissemination shall help in informing other members in the group about changes. After the discussion and agreement about ontology elements and execution of changes, dissemination of these ontology elements in the community is required to guarantee their usage, e.g. for the annotation of ressources. Tools like wikis (as proposed by [2]), or also the semi-automated search using text mining for links to web-pages (e.g. OntoGen [11]) describing these ontology elements and possibly the design rationale behind it are very helpful for that.

Awareness of changes also helps in controlling changes from other users. As indicated in [12], it is helpful to provide tools for taking over responsibility for them and promoting allegiance (e.g. for the creators of these ontology elements). Tools that allow users for the subscription for notifications to ontology elements, e.g. via e-mail, thereby

[3] `http://www.jforum.net`

help in notifying them in case of updates. In additions, it could also be helpful to offer tools that help in undoing changes identifyed as incorrect extensions of ontology elements – this is one of the instruments of Wikipedia for maturing support [13].

Detection Automated detection can help in finding unused and very similar ontology elements. In the evaluation of the imagenotion tool, there were ten imagenotions in the ontology with one work step each. This indicates unused and immature ontology elements. Therefore, it is helpful to offer tools for automated identification of candidates for cleansing of unused ontology elements to keep the collaborativly created ontology as compact as possible. In addition, it is also helpful to offer tools that help in marking ontology elements that are very similar. Then, it is possible to discuss whether they should be merged.

5.2 Support of Local/Private Data

Both Imagenotion and SOBOLEO currently support a very simple mode of sharing data: all data is shared globally and is jointly edited by everyone. Every statement created is owned by all users and can be seen, edited and deleted by every user. Imagenotion saves who created each statement, but it is not stored when multiple users have the same believe about annotating a resource. This model is similar to that of Wikipedia where a single version of each article is jointly maintained. A competing model is used by social bookmarking sites such as del.icio.us: here each user creates a personal view on the resources. The same tag used for the same resource by multiple persons is stored as two different statements. The personal views of multiple users are connected through the use of common tags.

In the evaluation of SOBOLEO users frequently complained about the lack of such a personal view. One comment representing this line of citique was: “provide a personalized citation browser – only show me the links that I added”.

We are currently working to support a combination of these two edit models for future versions of the two tools (also taking into account approaches like the HCOME methodology [14]): the Wikipedia model for the creation and maintenance of the shared vocabulary and the social bookmarking model for the management of the annotations. We also want to support the designation of parts of the shared vocabulary as uneditable; for example to ensure that the annotations created stay compatible with some standard vocabulary maintained elsewhere. Users should also be able to give different visibilities to the annotations, either public, private or visible to arbitrary user groups.

The infrastructure needed to support these use cases differs from well-known access control paradigms (e.g. in file systems) in two main areas: (1) the application of different rights to different parts of rdf-graphs is less well understood than the application of rights to strictly hierarchical data structures (2) the personal view is treated differently than privately editable data. The personal view can be understood as the utterances of a person; hence everyone can only edit her own utterances, but everyone is also free to repeat those of other people or even to make conflicting statemens. This is different from normal access control where private data is simply non-editable for others.

5.3 Communities or Groups and Perspectives

When an ontology is created collaboratively in a larger community, it can be assumed that it will quickly become unwieldy; i.e. that the ontology becomes to large to easily display it in editors, that one user cannot follow all ongoing discussions about changes, that most users are not able or willing to understand the details of parts of the ontology of little concern to them, that there are too many changes happening in quick succession etc. So far we have tried to avoid this problem by intentionally restricting the users to only a small group from a single domain trying to achieve a single joint goal. However, traces of this problem appeared even in our small-scale evaluation when some users started to create sophisticated conceptualizations of the world of military aircraft – to specialized to be of interest to the other users.

As a means to tackle these complexity, there is a strong case to allow for a kind of editable views on the global ontology – smaller ontologies or subgraphs of the ontology that users can commit to. These views could function like thematic user groups on sites like Flickr: e.g. a user interested in military aircraft would join a group specializing in this topic and would then be shown their view, could easily change the relevant concepts and the concepts from this group would be recommended during annotation. For search and browse activities all annotations and ontology would be used by the system, but a preference would be given to those from groups the user is member of. For example when locking at a picture that is annotated with a large number of concepts, a user would see the annotations created by her group(s) and a hint making her aware of other groups that have annotated this particular image. Through these hints she could navigate to the annotations of the other groups. In such a browse scenario, the display of the groups helps in grouping large numbers of annotations and also informs the user about the existence of other groups, thereby fostering consolidation between groups working on related topics.

Introducing such views, however, would come with considerable added complexity, both for the system and the user. At the one hand users would need to understand this added level of abstraction, must be shown and understand how the concepts in the ontology relate to the groups and understand what it means if they leave a group. At the level of the system there is the need not only to manage the groups and their views but also to further support users in finding synergies over groups and to support such complex operations as the merging of ontology created independently. In fact all four consolidation areas identified in section 5.1 apply on groups and views as well.

6 Conclusions and Outlook

Evaluations of our tools SOBOLEO and ImageNotion have confirmed that our ontology maturing approach is feasible to enable agile community-driven ontology engineering for communities of practice. While there are other proposed approaches like [15] sharing the same spirit, the focus on *work-integrated* ontology engineering has proven to be a crucial element, exemplified by our annotation use case.

But a more important result of these evaluation sessions was the guidance for further developments. We are aware that the key for success of ontology maturing support

is the right level of complexity: supporting needed actions while retaining ease of use. Therefore, it is crucial that we derive the future development route from actual user needs in a participatory design approach. From the first formative evaluation sessions, we have learnt that we need further developments in the following areas: (1) support for consolidation in all phases (candidate identification, discussion and agreement, execution and dissemination), (2) introduction of an individual scope as the possibility to have diverging private elements, and (3) support for different and diverging microtheories for specific communities/groups.

We will address these issues within our next iteration of development. We also plan to approach the problem of efficient ontology maturing support also in other use cases beyond annotation of resources within the FP7 Integrating Project MATURE[4].

References

1. Barker, K., Chaudhri, V.K., Chaw, S.Y., Clark, P., Fan, J., Israel, D., Mishra, S., Porter, B.W., Romero, P., Tecuci, D., Yeh, P.Z.: A question-answering system for AP Chemistry: Assessing KR&R technologies. In: Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning. (2004) 488–497
2. Hepp, M., Bachlechner, D., Siorpaes, K.: OntoWiki: community-driven ontology engineering and ontology usage based on Wikis. In: WikiSym '06: Proceedings of the 2006 international symposium on Wikis, New York, NY, USA, ACM Press (2006) 143–144
3. Braun, S., Schmidt, A., Zacharias, V.: Ontology maturing with lightweight collaborative ontology editing tools. In Gronau, N., ed.: 4th Conference Professional Knowledge Management - Experiences and Visions (WM '07), Potsdam. Volume 2., Berlin, GITO-Verlag (2007) 217–226
4. Braun, S., Schmidt, A., Walter, A., Nagypal, G., Zacharias, V.: Ontology maturing: a collaborative web 2.0 approach to ontology engineering. In: Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge at the 16th International World Wide Web Conference (WWW 07), Banff, Canada. (2007)
5. Hepp, M.: Possible ontologies: How reality constraints building relevant ontologies. IEEE Internet Computing **11** (2007) 90–96
6. Allert, H., Markannen, H., Richter, C.: Rethinking the Use of Ontologies in Learning. In Memmel, M., Burgos, D., eds.: Proceedings of the 2nd International Workshop on Learner-Oriented Knowledge Management and KM-Oriented Learning (LOKMOL 06), in conjunction with the First European Conference on Technology-Enhanced Learning (ECTEL 06). (2006) 115–125
7. Zacharias, V., Braun, S.: Soboleo - – social bookmarking and lighweight engineering of ontologies. In: Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge at 16th International World Wide Web Conference (WWW2007). (2007)
8. Brickley, D., Miles, A.: SKOS Core Vocabulary Specification. W3C working draft, W3C (2005)
9. Walter, A., Nagypal, G.: Imagenotion – collaborative semantic annotation of images and image parts and work integrated creation of ontologies. In: Proc. of 1st Conference on Social Semantic Web, Leipzig, Germany. (2007)
10. Konstantinos, K., Vouros, A., Stergiou, K.: Towards automatic merging of domain ontologies: The HCONE-merge approach. Journal of Web Semantics **4** (2006) 60–79

[4] `http://mature-ip.eu`

11. Fortuna, B., Grobelnik, M., Mladenic, D.: Semi-automatic Data-driven Ontology Construction System. In: Proc. of the 9th International multi-conference Information Society IS-2006, Ljubljana, Slovenia. (2006)
12. Maier, R., Schmidt, A.: Characterizing knowledge maturing: A conceptual process model for integrating e-learning and knowledge management. In Gronau, N., ed.: 4th Conference Professional Knowledge Management - Experiences and Visions (WM '07), Potsdam. Volume 1., Berlin, GITO (2007) 325–334
13. Braun, S., Schmidt, A.: Wikis as a technology fostering knowledge maturing: What we can learn from wikipedia. In: 7th International Conference on Knowledge Management (IKNOW '07), Special Track on Integrating Working and Learning in Business (IWL). (2007)
14. Kotis, K., Vouros, A.: Human-centered ontology engineering: The HCOME methodology. Knowledge and Information Systems **10** (2006) 109–131
15. Siorpaes, K., Hepp, M.: myOntology: The marriage of ontology engineering and collective intelligence. In: Bridging the Gep between Semantic Web and Web 2.0 (SemNet 2007). (2007) 127–138

Understanding and Supporting Ontology Evolution by Observing the WWW Conference

Nicolas Guelfi[1], Cédric Pruski[1,2], Chantal Reynaud[2]

[1] Laboratory for Advanced Software Systems – University of Luxembourg campus
Kirchberg 6, rue Coudenhove-Kalergi, L-1359 Luxembourg
[2] LRI – University of Paris-Sud, CNRS & INRIA-Futurs, 4 rue Jacques Monod, Parc Club
Orsay Université, 91893 Orsay Cedex (France)
{nicolas.guelfi, cedric.pruski}@uni.lu, chantal.reynaud@lri.fr

Abstract. Ontologies which represent domain knowledge in information systems are efficient to enhance information retrieval. However, domain knowledge is evolving over time and thus it should be also expressible at ontology level. Unfortunately, we consider that ontology evolution is barely study and its basic principles have not been yet precisely defined according to our notion of evolution. In this paper, we have followed a bottom-up approach consisting in a rigorous analysis of the evolution of a particular domain over a significant period of time (namely the WWW series of conference over a decade) to highlight concrete domain knowledge evolutions. We then have generalized and we present a precise set of evolution features that should be offered by ontology metamodels. We also evaluate the modelling capabilities of OWL to represent these features and finally, we show the contribution of ontology evolution support to improve Web information retrieval.

Keywords: Ontology Evolution, Domain Analysis, OWL, Web Information Retrieval

1 Introduction

Although being firstly introduced in philosophy, ontologies have recently appeared in the field of computer science as the cornerstone of the Semantic Web paradigm [3]. The later aims at giving a sense to the Web what will allow computers to "understand" Web data. If this goal is achieved, computers will be able to unload users of many tedious tasks like searching relevant documents or services. The Semantic Web implements ontology that models a part of the real human world, mainly to annotate Web data or to facilitate information retrieval. Nevertheless, since ontologies represent the knowledge of a particular domain, they have to smoothly follow the evolution of that domain otherwise their use will lead to unwanted effects. Therefore the ontology evolution problem [11], [15] has recently been deep studied since it becomes rapidly of utmost importance.

In our previous work we have defined the O^4 approach [6], [8] that aims at improving the results of a Web search in terms of relevance. This is done mainly through the use of ontology-based query expansion rules. In order to optimize the search, we need to select the adequate terms from the ontology to enrich the query. Actually, if the ontology does not reflect the knowledge of the domain associated to the submitted query, the search results will not be those awaited by users. We are thus facing the problem of ontology evolution.

In this paper we propose a set of modelling features for ontology evolution. These features have been defined after the rigorous study of the evolution of a particular domain (in our case, the domain defined by the WWW series of conference topics) over a ten years period of time. In consequence, we will first have to present in detail the construction of a corpus of documents that is representative of the domain we have studied. This requires the definition of relevant criterion and tools that will facilitate the analysis of the domain. The results of this analysis will lead directly to the definition of the various kind of evolution that can appear [7] which in turn will allow the proposition of modelling features that aims at designing evolving ontologies. The proposed primitives will first allow us to understand the evolution of ontologies and will aid to predict future versions of ontologies. They can be used to describe a structural evolution on one hand and a progressive evolution on the other hand. Since this work has been carried out in a context covering Web information retrieval, we will highlight the contribution of such ontologies through an example implementing ontology-based query expansion techniques [8] to improve the relevance of documents when searching the Web.

The remainder of the paper is structured as follows. Section 2 presents the characteristics of the domain we have studied in order to define the new modelling features devoted to ontology evolution. In Section 3 we detail the proposed modelling primitives as well as their properties. Section 4 illustrates an application of our work through a basic example dealing with Web information retrieval. In Section 5 we discuss related work in the ontology evolution field. Finally the paper wraps up with concluding remarks and our future work.

2 Domain of Study Definition and Ontologies Construction

The first step towards the proposition of modelling features devoted to ontology evolution concerns the construction of a significant corpus of documents that will allow us to highlight the various kinds of domain evolution. In this section we present the characteristics of such a corpus and the ontologies we have built from that pool of documents.

2.1 Domain Selection

Since we want to derive modelling features for ontology evolution from the analysis of the evolution of a particular domain, the selection of such domain is of utmost importance. Many domains, like bioinformatics through the Gene Ontology [16], are already modelled using ontologies. Unfortunately, these ontologies are either young

or built using only domain-specific relations. Therefore we considered that the study of their evolution will not be relevant enough and we decided to construct ontologies from a set of descriptions of an evolutionary domain. To this end, we have chosen the domain covered by the World Wide Web series of conference which is reflected in the calls for papers and in the accepted papers. Thus, papers accepted for publication at these events together with the calls for papers, which are online and can be retrieved via a web search engine, form our "Micro Web" (i.e. the corpus of documents we will analyze). In order to see the evolution of the domain of the Web over a significant period of time, we decided to harvest the accepted papers of the last 10 WWW events. The so-called *Micro Web* consists of good case study since the chosen conference is world famous and known to be one of the most representative events in the domain of the Web. Therefore, its successive calls for papers reflect the various fields in vogue in the corresponding domain. Moreover, the quality, quantity and homogeneity of the submitted papers as well as the high level of selectivity (less than 20%) set by reviewers reinforce this idea. As a result, we have a corpus made of 622 documents all stored in a relational database which will facilitate their future analysis.

2.2 Methodology for Ontology Construction

We have designed, from the calls for papers and the accepted papers of the different conferences, the ontology of the domain of the corresponding event for each year using the Protégé[1] ontology editor. It means that we have built 10 ontologies, one concerning each event of the WWW series. These various ontologies represent in fact the same domain that is evolving over time. The ontologies are constructed following a rigorous process inspired from the ARCHONTE methodology [2]. The three steps of this methodology consist in a semantic normalization of the terms introduced in the ontology, followed by a formalization of the meaning of the knowledge primitives obtained and an operationalization using knowledge representation languages. The so built ontologies will allow us to identify the different evolutions of the domain to try, in the next phase, to explain the changes. The construction of the different ontologies has been done manually following the process described hereafter. We first model the knowledge of the domain and then we formalized it using the Web Ontology Language (OWL). As modeling is the main purpose here, we use the most expressive flavor of OWL (i.e. OWL Full).

First of all, we stated that every topic of a call for papers denotes a concept in the corresponding ontology. This means that our ontologies are small and made of about 30 classes. For instance, the topic *multimedia* in 1998 provides the concept *multimedia* in 1998's ontology. Furthermore, topic like *social and cultural* gives rise to two concepts (i.e. a concept *social* and another one *cultural*) in the ontology. Indeed, we decided to split expressions according to the conjunction "and" which regularly appears. Nevertheless, the conjunction of words using this proposition indicates that words involved in that particular topic are linked. This is the first step towards the construction of the set of relations that bind concepts of the ontology.

[1] http://protege.stanford.edu/

Then, to determine these relations, we base on the content of the accepted papers with a particular attention devoted to the abstract of the papers. We first localize an occurrence of the concepts we tried to bind in a document of our corpus, and then we tried to identify manually from the text the relation of the domain. In order to validate our choice, we reiterated the operation on several other papers. This basic but rigorous process provides us the ontologies (one is depicted in figure 1) of the domain that will be the material of our study devoted to domain evolution.

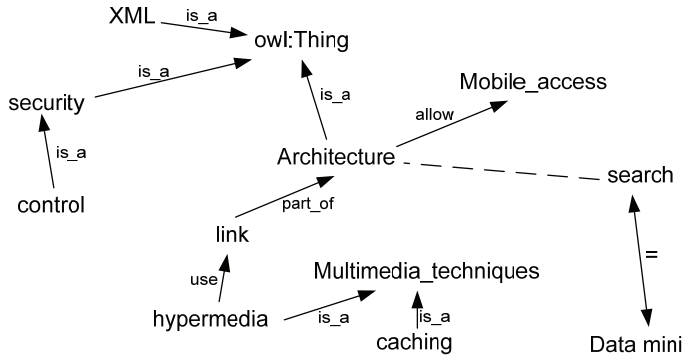


Fig. 1. Part of the Ontology representing WWW 2000's call for papers

As partially illustrated on the ontology in figure 1, we used very elementary relations like subsumption, equivalence or meronymy (i.e. part-of) to design our ontologies. Nevertheless, we also introduce some particular relations resulting from the analysis of the content of the papers like *use* or *allow*. All the constructed ontologies can be downloaded from our Web site[2].

3 Domain Analysis: Towards Ontology Evolution

The analysis of the evolution of the domain represented by WWW conferences' topics made it possible to define various kinds of evolution that affect the domain. In this section, we will detail these evolutions and as a result, we will give the corresponding modelling primitives for the design of evolving ontologies.

3.1 Domain Evolution

In this first subsection, we discuss the various kind of evolution that stand out during the analysis phase of our micro web. The analysis we made is at two levels. The first one, called general observation, defines the macroscopic evolution of the domain over a long period of time (10 years in our case). In contrary, the second level, called local observation, highlights the microscopic variation of the domain. This second kind of observations is made on a very short period (i.e. 1 or 2 years). These observations

[2] http://se2c.uni.lu/tiki/tiki-index.php?page=TargetTool

made it possible to emphasis different kinds of evolution among which one finds concept persistence, emergence of new concepts, concepts removal, generalization and specialization of concepts. However, our observations have also permitted to define other important features like the importance of concepts in the domain but also the resistance to modification and the variation of a distance between concepts over time as well as the speed of evolution. All these characteristics will be detailed in the remainder of this section. However, in order to explain the various highlighted kinds of evolution, we needed to distinguish between the ontology built from the calls for papers which represent the conferences chairman point of view and the content of the papers which represent the authors' interpretation of the calls. Unfortunately, we did not have access to the reviews. These would have permitted to understand if the authors' interpretations of the calls were consistent with the chairs point of view of the domain.

Concept Persistence

This first kind of evolution affects some particular concepts of the domain. Actually, we observe that special concepts like *security* or *search* are present in the ontology over the whole period of observation. It means that since its appearance in the domain, the concept remains in the domain. We called this constraint on evolution **concept persistence**. Our personal knowledge of the domain lets us claim that these two concepts denote key notions of the domain (recall that we study the domain covered by World Wide Web series of conference). Thus, we can say that concepts that are part of the ontology over a predefined long period of time constitute the core of the ontology because the semantics of the concept is still covered by the semantics of the domain. This is particularity important for approach exploiting ontologies like techniques for indexing data or data retrieval. In fact, these concepts are the most relevant and in consequence should be favoured in their usage. For instance, the *search* concept is present in the domain for the whole period of study, whereas other concepts which seem to be less important like *social* remains in the domain only for one year. We will illustrate this particular point in Section 4 hereafter.

Concept Emergence

The second observation in the evolution of a domain concerns the addition of knowledge at a particular moment. This **emergence of concept** was particularly true for the *Semantic Web* in 2002. Since this paradigm was defined in 2001 by Tim Berners-Lee [3], and its associated semantics was close enough from the semantics covered by the domain defined by the topics of the WWW series of conference, it rapidly appeared as a concept of the domain and as a result, one year later in the topics of the WWW conference. This is why it takes place in our ontology representing the domain covered by WWW 2002 topics. Our survey has shown that 79 concepts have emerged in the domain of the WWW series of conference between 1997 and 2007. Moreover, there are about 11 concepts in average that emerge each year in an ontology that contains about 30 concepts. Recall that we have one ontology per conference.

Concept Removal
A concept can be removed from a domain for several reasons. The first one is related to its semantics. Actually, by virtue of knowledge evolution, the semantics of that concept could not be covered by the semantics of the domain (described by the ontology) anymore and therefore should be removed from that domain. Moreover, a concept can be either not precise enough (i.e. the concept is too abstract) or too precise (i.e. specific concept). This would also require some domain refinement which in turns will lead to the removal of concepts for the benefit of more abstract or specific concepts. Another reason concerns the properties of the concept. For instance, if the concept is no more "popular" or "profitable" (if we are in a business domain) for the domain it can also be removed. We can speak about obsolete concept. In our case study this kind of evolution arose several times. For instance, concepts like *social* and *cultural* appear in the 1998 WWW conference topics but are removed in the 1999 conference topics and does not appear anymore. Moreover, our study revealed that in the WWW 1998 conference, no papers containing these two words were submitted which proves a kind of irrelevance. This is probably the reason why both concepts have been removed from the domain from that moment.

Concept Abstraction
Our observations revealed that a concept or several concepts can be substituted along time axis by a more general concept. We call this phenomenon **concept abstraction**. This can be done when the semantics of a concept is completely covered by the semantics of a concept that is directly link to it. However, we observe that this phenomenon usually turns up when a concept is becoming less relevant for the domain. For instance, in our ontologies concepts like *browser* and *tool* are generalized into the more general concept *application*. This substitution give less importance to the two concepts that have been generalized which in turns give more freedom for the future authors in their interpretation of the call for papers. Actually, since this evolution in the call for papers took place, there have been more submitted papers dealing with a wider range of applications than papers discussing only the use of Web browsers. Our study has permitted to emphasis this idea. Concretely, there are 351 occurrences of the word *browser*, 173 occurrences of the word *tool* and 351 occurrences of the term *application* in the documents. Moreover, 30% of these words are cited in the same papers and in most of the cases, the words *browser* and *tool* can be replaced by the term *application* without a loss of semantics (i.e. the sentences where this phenomenon appears have the same meaning after terms substitution). Therefore the concepts of the ontology representing these notions (i.e. *browser* and *tool*) have been substituted by a more general one (i.e. *application*) which gave place to a wider variety of papers on Web applications. We have identified 5 concepts that have been abstracted. However, the time needed for a concept to become more abstract varies. Actually, some abstractions are very fast, only one year for the abstraction of concepts like *browser* and *tool*, other highlighted abstractions can be longer.

Concept Specialization

In the contrary, our empirical study has shown that a concept or a group of concepts can evolve in a more **specific concept**. Contrary to concept abstraction presented above, this phenomenon is possible only if the more specific concept on one hand shares a part of the semantics of its super concept and on the other hand, offers some specifics axioms that make it possible to represent the domain (or the subpart of it) at the right level of abstraction. In this particular case (i.e. concept specialization), the main objective is to bring more precision in the description of the domain by introducing new concepts. In our domain of study, to know the Web, this has been the case for the concepts *language*, *programming languages*, *markup language* and *metadata system* in 1998. Indeed, they have been transformed in a more specific concept: *XML* the year after. This modification that brought more precision in the call for papers has have an important impact on the submitted papers since 23 papers dealing with XML have been accepted in 1999. However, this rapidity in the change (only one year) can be explained by the analysis of the content of the papers submitted in 1998. We first observe that the *XML* word appears mostly in the paper of the track corresponding to the concepts that have changed (*language, programming language* …). Concretely, there are 162 occurrences of the term *XML* in papers related to programming languages, metadata systems and markup languages for a total of 205 occurrences of *XML* in all the accepted papers. Furthermore, the study of the abstract of these papers has highlighted that the concept *XML* was directly linked to the concepts *metadata, languages and markup languages* through a subsumption relation and terms *metadata, markup language*, and *programming languages* refer in most of the case to *XML* in the content of the papers. This phenomenon combined with the relevance of the XML language at this period of time has probably egg WWW 1999 chairman to adapt the call for papers. This observation underlines the relations between the interpretation of the domain (i.e. the content of the papers) and the evolution of the domain itself (i.e. the call for papers). 7 concepts have been specialized over the period of study and this evolution gave place to 16 new concepts of the domain. Moreover, as it is the case for abstraction, the operation required more or less time depending of the nature and the importance of the concept in the studied domain.

Semantic Weight

Another important kind of evolution that has been highlighted by our study deals with the notion of importance of the concepts in the domain. We call this phenomenon **concept emphasis.** This property put the stress on the punctual tendency of the evolution. In fact, at some time, concepts are more relevant for the domain than other ones. Depending on the domain of interest, this "relevance" can be popularity, profitability, technological improvements, etc. In our study, this is the case for concepts like *search*, *hypermedia* or *Semantic Web* in 2002 but also *ontologies* recently in 2006. This turns up at two different levels. First, it appears in the tracks of the conference. In fact, since there have been so many accepted papers dealing with these notions, two tracks were organized which underlines the importance or the **semantic weight** assigned to these topics. Second, 83% of the accepted papers of the other tracks contain at least one occurrence of the involved word which is also an indication concerning the important aspect of the concept *ontology* in the domain. We

believe that this notion is really important and we will give an illustration in Section 4.

Semantic Distance
A more meticulous observation of the evolution of the domain of the Web through the calls for paper of WWW's events has permitted to emphasize the notion of **semantic distance** between the concepts of the ontology. However, the distance we highlighted is different from those proposed by Hirst-St-Onge [9], Jiang-Conrath [10] or Resnik [14]. Actually, these metrics measure the distance between concepts that are linked by at least a path composed by more than one arc in the graph of an ontology and the objective is to estimate the closeness given their localization in the graph and the number of arcs that separate them in this ontology. Nevertheless, we found, through our empirical study, that the distance between concepts directly linked by the same arc in the graph of the ontology varies. Actually, some concepts seem to be "closer" (from the semantic point of view) than other ones although they are linked by the same relation in the ontology. This turns up in the use of the words denoting these concepts in the documents of the corpus. For instance concepts *browser* and *application* appear more frequently in the papers than concepts *tool* and *application* in 1999 and both couple of concepts are bounded by the same relation (in this case the relation of subsumption). Nevertheless, adequate metrics (different from those cited in this subsection) are needed to catch this notion of semantic distance. For the time being, we decide to consider words frequency. It means that we measure how many times two concepts of a relation are cited together in the same kind of documents (i.e. documents published the same year) and in the same context. Moreover, this distance plays a key part to explain for instance the removal of concepts from the ontology. In fact, concepts which are not relevant anymore for the domain, are getting further and further form other concepts of the domain (i.e. the semantic distance is increasing). Therefore, when a predefined threshold is reached, concepts are removed from the domain. In the contrary, when concepts are very close, they can be replaced by a more abstracted or specific concept if another appropriate threshold is reached.

Resistance
This other kind of evolution, called **resistance to change**, is a bit different from the other characteristics presented so far. Actually, it has the particularity to be opposed to evolution. This appears in our study in the ontology of 1998 and 1999. It reflects also in the documents of the corpus. Indeed, there are 49 occurrences of the words security in the papers accepted in 1999 which is very few. Furthermore, one paper contains 26 occurrences of that word. This reveals that the notion of security was not of utmost importance in 1998. Thus, following the natural aspect of the evolution process, this concept should have been removed from the ontology representing WWW 1999's call for papers which is not the case as the concept security remained in the call for papers in 1999. This proves that the chairman of WWW 1999 has considered this notion as important for the field. The resistance to changes is also present in other field mainly knowledge management [4], [12]. Nevertheless, the resistance seems to vary according the concepts involved. Each concept resists differently to evolution. The "coefficient" of resistance to change affected to each

concept is different. This introduces a notion of degree of freedom in the evolution of the ontology. In fact, using this property, one can partially control the evolution of the ontology. Thus, this newly introduced metrics should be determined rigorously by domain experts. Furthermore, this phenomenon turns up under various forms in every day's life. For instance, for approximately 80% of the population, whales are seen as fish and for only 20% of the people whales are mammals. In consequence, if we follow the natural evolution process, in a significant period of time, all the people should classified whales under fishes. Nevertheless, among the 20% of the people are biologists (i.e. domain experts) that will permanently reject this evolution. This proves the existence of such resistance to change and should be taken into account in the ontology representing the domain mainly using adapted coefficient as shown in section 3.2.

Speed of Change
The evolution of the domain takes place at different speeds. Some changes are rapid; others are very slow and required several years. For instance, the specialization concepts *metadata systems, programming languages, and markup languages* into *XML* (as presented before) has taken only one year in the contrary, concepts *browser* and *tools* have been abstracted into *application* in 2 years. We believe that the speed of change is function of the coefficient of resistance to changes presented before. In fact, if the coefficient is high, it means that the ontology should not change (or change very little) which ensures a kind of stability in the ontology. However, if the same coefficient is low, it will allow more flexibility in the evolution of the ontology. The speed of change depends also on external factors like technology improvements. This was the case for the concept *Semantic Web;* only one year after its definition it became a key concept of the domain.

3.2 Modeling Features for Ontology Evolution

The various kinds of evolution we have highlighted through our empirical study, have led us to the proposition of modeling primitives for ontology evolution. In this section we describe these various modeling elements.

The proposed features can be classified into two different sets. Actually, we have primitives that act on concepts (i.e. vertices of the ontology graph) and also primitives that apply on relations (i.e. edges of that graph). The first set is made up of primitives that put the stress on concepts emergence, concepts persistence and concepts importance. So, first when a new concept emerge in a domain, it is important to know the exact date at which the concept has appeared in the ontology. Second, concerning persistence, two things are needed. On one hand, the **emergence date** and on the other hand a **duration** determined manually by domain experts. The latter correspond to a constraint of time the concept has to satisfy in order to be considered as persistent. The last modeling feature that applies on concepts is related to concept importance. We have decided to model this property using a coefficient called **importance**. For the time being, this coefficient is computed based on the occurrences and the repartition of the given concept in our corpus of documents. In consequence, the more frequent its associated term is cited and the better the

repartition of this term in the corpus is, the higher the coefficient of importance will be. We decided to limit the coefficient between 0 and 1 (1 representing a very important concept).

The second set is formed by modeling primitives affecting relations between concepts. These are related to the semantic distance and the resistance to changes. Both notions are represented using **coefficients**. The semantic distance between two concepts measures the evolution of the joint use of these two concepts in the corpus of documents. This coefficient can vary from 0 to infinite but a maximum distance is set by domain experts and if the distance reaches this particular value, the relation between the two concepts is removed (i.e. an edge of the ontology graph is removed). Moreover, if one concept becomes isolated in the ontology (i.e. it is no more linked to any other concepts) it can be removed from the ontology. Concerning the coefficient of resistance to changes, it must be defined by domain experts. This coefficient takes its values between 0 and 1 where 1 denotes a very strong coefficient which prevents every relation that is affected to evolve.

As OWL is the *de facto* standard for designing ontologies, we decided to study how to represent the modeling elements presented in this paper using the OWL language. Due to its powerful expressivity, OWL offers enough characteristics to take all the presented features into account. Table 1 hereafter presents the various modeling features, their associated datatype and the ontology notions they are applied to. Observe that the types we use are the same than those contained in XML schema definition.

Table 1. Modeling Features Summary

Element Name	Datatype	Affect
Emergence date	xsd:dateTime	concept
Persistence	xsd:duration	concept
Importance	xsd:float	concept
Semantic Distance	xsd:nonNegativeInteger	relation
Resistance	xsd:float	relation

However, OWL metamodel[3] [11 p.83] should be enriched in order to integrate the above modeling features as basic OWL primitives. Concerning all features that apply on concepts, three attributes should be added to the class *Class* of the OWL metamodel. One attribute for representing the emergence date of a concept in the ontology, a second one to express the persistence duration of a concept and finally a third one for the importance of a concept. Moreover, these attributes must have the same type than their associated elements (see table 1).

The two modeling elements related to relations, can be integrated to the OWL metamodel by adding two attributes to the class *Property.* A first non negative integer concerning the semantic distance and a second float for expressing the notion of resistance to changes are needed. Nevertheless, the expressivity of OWL makes it possible to easily express properties concerning concepts of an ontology without

[3] The OWL metamodel we refer to is the one described using UML by Klein in his PhD Thesis.

modifying the OWL metamodel mainly using OWL Datatype properties but for elements related to OWL properties it would be more problematic.

Another way to proceed would consist in using annotation properties or datatype properties via datatypes defined in accordance with XML Schema datatypes to express our concepts at ontology level. Nevertheless, this would require the expressivity of OWL Full.

4 An Application to Web Information Retrieval

In this section we describe a real contribution of adaptive ontologies in the context of information retrieval. Our formerly mentioned O^4 approach [6], [8] implements an ontology-based query expansion technique in order to improve the results, in terms of relevance, when searching the Web. Actually, the query expansion phase is made according rigorous expansion rules defined by taking into account terms of the query, the form of the initial query and the relations that link the concepts of the query in the ontology. The ontological relations implemented in this approach are on one hand the well-known equivalence and subsumption relations which are already implemented in OWL and on the other hand part-of and opposition relations which have all been formalized in first-order logic and added to the Web Ontology Language as primitives [8]. A first basic rule consists, given a basic query constituted by only one keyword, in adding all the equivalent concepts of this keyword in the ontology. Nevertheless, the ontologies we implemented so far were not able to evolve over time and thus do not reflect the knowledge evolution of the domain they model. In consequence, the choice of the right terms to put in the query was not fine enough. Due to the properties of the evolution features we have presented in this paper, and mainly the semantic distance and the semantic weight assigned to concepts of the ontology, we will be able to refine even more this choice by selecting concepts which weights are the highest since they are considered as the most relevant concepts of the domain. The results of such a search will be more relevant because the more relevant concepts of the domain will be added to the query.

Assume to illustrate this argument that an initial query “Web” will be submitted to a Web search engine. If, for instance, the ontology we use to perform query expansion contains two equivalent concepts for Web that are “WWW” and “Internet” with a semantic distance from “Web” of 1 and 10 respectively. The system will select the term “WWW” to put in the query since it is closer to the initial term “Web” than “Internet” is close to “Web”. So, the expanded query “Web WWW” will be submitted. Such expansion is judicious if we compare the different search results associated to both queries “Web WWW” and “Web Internet”. Actually, pages returned when the query “Web Internet” is entered are older and probably out of date than pages returned corresponding to the other query. That shows that the integration of domain evolution at ontology level will improve Web information retrieval at least by giving right up to date information. Another basic example consists in filtering the returned pages using the emergence date and the persistence duration of concepts that constitute the query. Assume that the query “modem Internet” is submitted to a Web search engine. The system would be able to return pages dealing with modems that

were published from the emergence date of *modem* in the domain of the *Internet* and for the persistence duration of the *modem* concept.

This is all the more true for approaches implementing ontologies for tagging or indexing information. Since the vocabulary for indexing or tagging is extracted from ontologies, it has to be selected rigorously. Moreover, tags are usually chosen by taking their popularity or any other properties that is domain dependent into account. However, this kind of information was not provided by static ontologies. Nevertheless, we have proposed an approach that has the advantage to integrate such properties directly at ontology level. Therefore, if the concepts presented in this paper will be integrated directly in ontologies, they will have a huge impact on approaches dealing with tagging or information indexing.

5 Related Work

In the field of ontology evolution, relevant work has been carried out but two main different approaches stand out. The first one, inspired by the work done in the database field, considers ontology versioning. This problem has mainly been tackled by Michel Klein [11]. He compared ontology evolution with database schema evolution. The framework he proposed contains a set of operators, on the form of an ontology, useful for modifying another evolving ontology. Klein also proposes a change specification language based on the ontology of change operations. Moreover, Avery and Yearwood, through their extension of OWL called dOWL [1], have proposed a set of primitives to improve ontology versioning by facilitating the design of dynamic ontologies.

The second approach for ontology evolution deals with consistency during the evolution process. To this end, Ljiljana Stojanovic proposed a general methodology for managing ontology evolution [15]. The process can be divided in 6 different phases occurring in a cyclic loop. It enables handling the required ontology changes; ensures the consistency of the underlying ontology and all dependent artifacts; supports the user to manage changes more easily; and offers advice to the user for continual ontology reengineering. Recently, Peter Plessers [13] described another framework for managing consistent changes in ontology. This is done through the definition of a *Change Definition Language* and the notion of *version log*. The former is a temporal logic based language that allows ontology engineers to formally define changes whereas the latter stores for each concept ever defined in an ontology the different versions it passes through during its life cycle.

Besides, another interesting work has been carried out by Giorgios Flouris [5]. It consists in applying approaches related to *belief change* to the ontology evolution problem. The set of modeling features we propose introduces a new dimension in ontology mainly by the introduction of the *Semantic Distance* between concepts of the ontology. Nevertheless, our approach is different from the two approaches existing in the literature which are reviewed in this section to know ontology versioning and ontology evolution management. In our approach, we represent the knowledge related to domain evolution in an ontology and show how this knowledge can be exploited in Information Retrieval. Moreover, these new properties will allow

first to understand the evolution and will make it possible to anticipate future evolution. Nevertheless, the dynamic ontologies we obtain can support ontology versioning and moreover, since we formalized our ontologies in OWL, techniques for change management can be applied too.

6 Conclusion

In this paper we have presented a domain analysis over a significant period of time leading to a set of ontologies corresponding to the same views of a same domain over different periods. We analyzed this set of ontologies in order to define new modelling elements dealing with ontology evolution. Moreover, we also illustrate the potential contribution of our proposition through an example dealing with information retrieval. We believe that the evolution features we have defined consist in an important step towards automatic ontology evolution. This will be possible if we find a way to analyze the corpus of documents automatically. Nevertheless, our approach needs to be strengthened mainly through the proposition of good metrics that will be able to characterize as faithfully as possible the status of knowledge in a corpus of documents from an evolution point of view. Therefore, our future work will concern on one hand the definition of such metrics and on the other hand, the proposition of a formal set of operators able to, given a corpus of documents, update automatically the appropriate elements of the ontology we have introduced in this paper.

References

1. Avery, J., Yearwood, J.: dOWL: A Dynamic Ontology Language. In: Proceedings of the IADIS International Conference WWW/Internet 2003, Algarve, Portugal, IADIS (2003) 985-988
2. Bachimont, B., Isaac, A., Troncy, R.: Semantic Commitment for Designing Ontologies: A Proposal. In: 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'02). Volume LNAI 2473., Sigüenza, Spain (2002) 114-1213.
3. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American **284**(5) (2001) 34-43
4. Biscaccianti, A., Renard, P.: The Cooperative Contextual Change Model: a Systemic Approach to Implement Change while Preserving Stability. Cahiers du CEREN **4** (2003) 1-16
5. Flouris, G.: On Belief Change and Ontology Evolution. PhD thesis, University of Crete, Heraklion (2006)
6. Guelfi, N., Pruski, C.: On the use of Ontologies for an Optimal Representation and Exploration of the Web. Journal of Digital Information Management (JDIM) **4**(3) (2006)
7. Guelfi, N., Pruski, C., Reynaud, C.: Towards the Adaptive Web using Metadata Evolution. In Calero, C., Moraga, M.Á., Piattini, M., eds.: Handbook of research on Web information systems quality. Idea Group Publishing (2007)
8. Guelfi, N., Pruski, C., Reynaud, C.: Les ontologies pour la recherche ciblée d'information sur le web: une utilisation et extension d'owl pour l'expansion de requêtes. In: Proceedings of the Ingenierie des Connaissances 2007 (IC07) french conference, Grenoble (July 2007)

9. Hirst, G., St-Onge, D.: Lexical Chains as Representation of Context for the Detection and Correction Malapropisms. In Fellbaum, C., ed.: WordNet: An electronic lexical database and some of its applications, Cambrige, MA, The MIT Press (1998) 305-332
10. Jiang, J., Conrath, D.: Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. In: Proceedings on International Conference on Research in Computational Linguistics, Tapei, Taiwan: Academia Sinica (1997) 19-33
11. Klein, M.: Change Management for Distributed Ontologies. PhD thesis, Vrije Universiteit Amsterdam (2004)
12. Maurer, R.: Beyond the Wall of Resistance: Unconventional strategies that build support for change. Bard Press (1996)
13. Plessers, P.: An Approach to Web-based Ontology Evolution. PhD thesis, Vrije Universiteit Brussel (2006)
14. Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: IJCAI. (1995) 448-453
15. Stojanovic, L.: Methods and Tools for Ontology Evolution. PhD thesis, University of Karlsruhe, Universität Karlsruhe (TH), Institut AIFB, D-76128 Karlsruhe (2004)
16. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. The Gene Ontology consortium. Nat Genet **25**(1) (May 2000) 25-29

A Framework for Cooperative Ontology Construction Based on Dependency Management of Modules

Kouji Kozaki, Eiichi Sunagawa, Yoshinobu Kitamura and Riichiro Mizoguchi

The Institute of Scientific and Industrial Research (ISIR), Osaka University
8-1 Mihogaoka, Ibaraki, Osaka, 567-0047 Japan
{kozaki, sunagawa, kita, miz}@ei.sanken.osaka-u.ac.jp

Abstract. To construct large scale ontologies, two major approaches are discussed by many researchers. One is a cooperative construction of ontologies, and the other is a modularization of ontologies. To combine these two approaches, this paper discusses a framework for supporting cooperative ontology construction based on dependency management among modularized ontologies. In such a situation, one of the key issues is the maintenance of consistency among inter-dependent ontologies because each ontology is revised asynchronously by different developers. In order to realize consistent development of ontologies, the framework provides two functions: to manage the dependencies between ontology modules and to keep and restore consistencies between them when they are influenced by changes of other modules. Furthermore, we outline an implementation of our framework in our environment for building/using ontology: Hozo.

Keywords: Cooperative ontology construction, Distributed development, Dependency management

1 Introduction

Ontological engineering has changed considerably for these years. Many systems have become to deal with multiple and dynamic ontologies rather than single and static ones. This trend is and will be accelerating because of the advancement of Semantic Web whose characteristic is decentralized. On the web, ontologies will be scattered from server to server and referred to by one another. For example, ontology creators and service providers would search and compile several ontologies on the web, and then, adapt them to their own needs. Especially, to construct large scale ontologies efficiently, many researchers discuss a modularization of ontologies [1]. Such modularized ontologies are treated meaningfully in every phase of the development process. At the beginning of ontology development, developers need to determine the scope of the ontology, and next, consider reuse of existing ontologies [2]. In these phases, dividing the target ontology into modules helps the developers to understand a total picture of the conceptual hierarchy particularly in a large scale ontology. And, it also helps to determine the scope of application of the reused ontology. In a phase of construction and maintenance, it forms the basis of

cooperative development. Furthermore, after publication of the ontologies, a developer of another ontology can reuse them as his/her own modules easily without carving out them from their source ontology if it is divided into modules in a reasonable manner.

In this research, we focus on the phase of construction and maintenance and discuss a framework for supporting cooperative ontology construction based on modularization of ontologies in a distributed environment. In such a situation, one of the key issues is the maintenance of consistency among inter-dependent ontologies because each ontology is revised asynchronously by different developers. In order to realize consistent development of ontologies, the framework has to support two functions: to manage the dependencies between ontologies and to keep and restore consistencies of them when they are changed[3]. This paper overviews the framework for distributed and cooperative ontology development based on dependency management of modularized ontologies and explains how the framework supports to keep and restore consistencies of the modules in the development processes. In this work, we have reconsidered the prototype system in previous work and improved its implementation. The remainder of this paper is organized as follows. Section 2 discusses the underlying distributed and cooperative ontology development we assume. In section 3, we summarize a flow of the distributed and cooperative ontology construction and discuss how to support each construction process in our framework. Section 4 introduces the implementation of our framework in our environment for building/using ontology: Hozo. In section 5, we discuss some related work followed by a summary of future work in section 6.

2 Distributed and Cooperative Construction of Ontology

We assume a situation where several modularized ontologies are constructed separately in a distributed environment and in parallel by different developers. In such a situation, some ontologies may import concepts (classes) defined in other ontologies, and another concept might be defined in the ontology by extending the imported concepts (Fig.1). And then, it means the ontology B which imports concepts from Ontology A depends on ontology A. In this paper, we call ontologies which are depended by other ontology and those depend on others *depended ontologies*, and *dependent ontologies*, respectively. In Fig.1, Ontology A is the depended ontology of Ontology B, and Ontology B is the dependent ontology of A. We call a development of ontologies in such a manner distributed ontology development.

In the distributed ontology development, developers construct multiple ontology modules in cooperation among the developers. They can reuse published modules of other ontologies if possible. It is a common way for ontology development to import an existing ontology into a target-specific ontology. However, when developers construct ontologies in parallel or reuse ontology which is under construction and thus unstable, consistency among the ontologies is easily broken because they are revised asynchronously without notice. Furthermore, they are sometimes updated without considering how ontologies depend on them would be influenced by their changes because authorities for maintenance of the ontologies are separated and distributed to

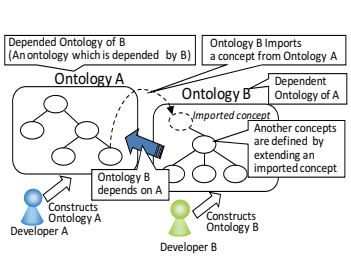


Fig.1. Distributed Ontology Development.

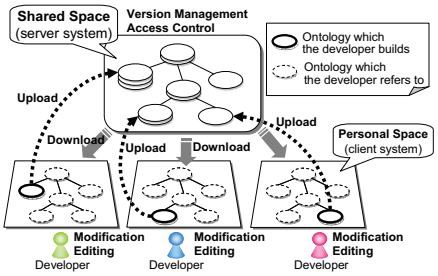


Fig.2. A Conceptual Framework for Distributed Ontology Development.

each developer. Therefore, when a developer changes his/her ontology, the change influences on its dependent ontologies. In many cases[1], such a change may cause inconsistencies among the ontologies. For consistent development of ontology modules, a system should manage dependencies among them and support their developers to harmonize them. Based on this observation, we have investigated how to manage the dependency among ontology modules and how a change of one ontology influences on others through its dependencies. And we have developed a framework for cooperative ontology construction in harmony among depended/dependent ontologies. Next section discusses the framework.

3. A Framework for Cooperative Ontology Construction based on Dependency Management of Modules

3.1. Flow of Distributed Ontology Development

Fig.2 shows a skeleton of our conceptual framework for distributed ontology development. It consists of two parts in a server-client architecture. One is a shared space, where developers store ontologies to be open to other developers. The other is local (personal) spaces, where each developer builds and modifies each ontology which he is responsible for. The developers cannot edit the ontologies stored in the shared space directly. Under access control and version management, they edit the personal copies of ontologies locally and upload them to the shared space when necessary. In the distributed ontology development, the target ontology can be regarded as a system of interrelated ontology modules stored in the shared space. They are constructed in cooperation among the developers. Each developer constructs some of them under his responsibility[2]. Then, he may refer to other ontologies and import concepts defined in them. It implies that each developer has two kinds of ontologies: ontologies which the developer builds and ontologies which he/she refers

[1] We assume early stage of ontology development by trial and error.
[2] The same component ontology may be constructed by several developers asynchronously.

to. The distributed ontology development proceeds with the repetition of the following steps;

1. A developer gets latest information on ontologies which he builds or refers to from the ontology server. He downloads (updates) them from the shared space to the personal space (client) through an ontology manager. If it is needed, he locks ontologies to avoid simultaneous modification of the same ontology by others.
2. The developer analyzes changes in the updated ontologies and evaluates whether the changes are influencing on consistency of the ontology which he is constructing.
3. If the changes cause inconsistency in his ontology, the developer modifies his ontology in order to keep and restore its consistency with the updated ontologies. The framework helps such a modification process by suggesting possible countermeasures for coping with each of the changes.
4. After the modification, the developer starts editing his ontology as he needs. While editing the ontology, he can imports and use concepts from other ontologies which he refers to as a result. Then the dependency between his ontology and the referred ontology through the imported concepts is managed by the functions of dependency management.
5. After editing, the developer publishes his ontology by uploading (committing) it to the shared space. Then, he unlocks the ontology if he allows others to edit it.

Every developer goes over the above process individually in parallel, and then the whole target ontology evolves. As a result the whole target ontology is constructed in the shared space.

We suppose another cooperative development process such as constructing a single ontology by many developers. Our distributed ontology development also can support such a process in the repetition of the following steps:

1. The developers share a target single ontology in the shared space. The ontology server manages versions of the ontology and accesses to it.
2. When a developer edits the target ontology, he locks the ontology and downloads (updates) it to his personal space.
3. If the ontology has been updated by another developer, he analyses the changes by comparing the ontology with its old versions. The change analysis function of the framework supports him by showing the changes and its influence.
4. After the analysis, the developer edits the ontology. And then, he uploads (commits) the edited ontology to shared space and unlocks it.

To support the distributed and cooperative ontology construction discussed above, the framework provides four functions:(1)sharing ontologies on the shared space under version management and access control, (2)dependency management among modularized ontologies, (3)analysis of changes and their influences, and (4)suggestion of possible countermeasures for coping with each of the changes to keep and restore consistencies. We discuss the details of these functions in the following sections.

3.2. Version Management and Access Control of Ontologies

As a basic infrastructure for cooperative ontology construction, our framework provides the following two functions:

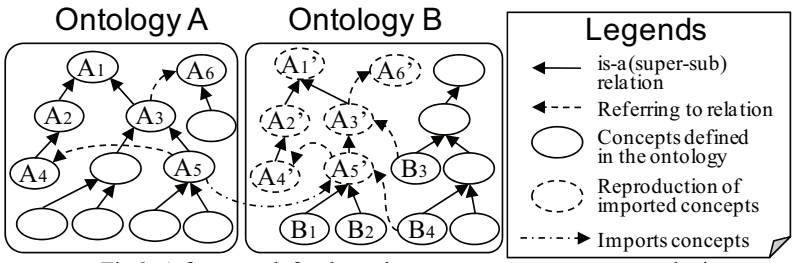


Fig.3. A framework for dependency management among ontologies.

Version Management: When the user uploads his ontology on the shared space (server), the old version of the ontology is moved to a backup space in the server. It is managed with its updated time and name of the developer, and it may be replaced by the latest version when the user requires.

Access Control: The server provides a mechanism for locking / unlocking ontologies to avoid that an ontology is updated by different developers at the same time. Because the units to be locked are modularized ontologies, its influence on the cooperative construction is kept to a minimum.

3.3. Dependency Management among Ontologies

When an ontology imports concepts[3] from other ontologies, the dependencies among ontologies are managed using reproduction of the concepts to be imported. As an example, we assume Ontology B imports concept A_5 defined in Ontology A (Fig.3). Then all the concepts depended by A_5 are reproduced with relations among them, and Ontology B imports these reproductions. It means the system reproduce all definition[4] related to the concept. In this example, "the super concept of A_5" (A_3 and A_1), "the concept referred by A_5" (A_4), "the super concepts and referred concepts of them (A_1, A_3 and A_4)" (A_1, A_2 and A_6) and relations among them are reproduced, and Ontology B imports these reproductions (they are shown by A_1' to A_6'). As the result, Ontology A becomes the depended ontology of Ontology B, and Ontology B becomes the dependent ontology of A. These reproductions have same definition with their original but belong to dependent ontology.

In Ontology B, another concept might be defined in the ontology by extending the imported concepts. The ways are divided into two types: defining sub concepts of them (B_1 and B_2 in Fig.3) and referring to them as constraints (B_3 and B_4 in Fig.3). These two types are represented by *is-a* (super-sub) relations and *referring-to* relations between reproductions of imported concepts and concepts defined in the ontology. These reproductions are used to manage dependencies among ontologies and to identify changes of depended ontologies. Because the dependencies are managed using relations between imported concept and concepts defined in dependent ontology [3], multiple dependencies (e.g. A depends on B, and B depends

[3] In OWL, the users cannot import a single concept, but they can import a whole ontology. But in our framework, the users may import concepts partially.

[4] The definition of concepts consists of id, name, super concept, comment, and slots.

on C) and circular dependencies (e.g. A depends on B, and B depends on A) can be managed by this framework.

3.4. Analysis of Changes and Their Influences

When a depended ontology is changed, the changes are analyzed by comparing its reproductions of imported concepts in the dependent ontology and their original concepts in the depended ontology. The types of changes are as follows:

1. If the original concept is not found[5] in the depended ontology, it means the concept was deleted.
2. If the definition of the original concept is different from that in the reproduction, it means the concept was modified.

The influences of the changes are analyzed by tracing the relations of reproductions whose original concept is changed. In Fig.3, we assume A_2 in Ontology A has been deleted. It means original concept of A_2' in Ontology B has been changed, and the change influences on A_4', A_5', B_1, B_2 and B_4 through their relations.

This analysis procedure is applicable to analyze the difference of an ontology and its old version. In the case, the comparison is done through all concepts and relations in the ontology. And the types of changes are as follows:

1. If a concept/relation is found only in the new ontology, it means the concept/relation was added in the new ontology.
2. If a concept/relation is found only in the old version, it means the concept/relation was deleted in the new ontology.
3. If the definition of a concept/relation in the new ontology is different from the same concept/relation in the old version, it means the concept/relation was modified.

The users can cancel part of the changes if it is needed.

3.5. Maintenance of Dependencies among Ontology Modules

For prevention and resolution of inconsistency in dependencies between ontologies, we can consider two approaches to maintain the consistencies. One is to restrict the change which influences on others seriously. Such a restriction helps developers to avoid inconsistency proactively. The other approach is to adapt the influences of the change and restore the consistencies by modifying influenced ontologies. We have taken the latter approach and have come up with five kinds of countermeasures for coping with each of the changes to keep and restore consistencies:

1) To accept the change

- **1-1) To modify the influenced ontology to be compliant with the change;** The developer makes agreement on the change of the ontology and modifies his/her ontology depending on it for adapting to the changed ontology.
- **1-2) To leave the depending ontology influenced by the change;** In some cases, the influenced ontology can be left unmodified, as the changed ontology does not contradict it.

[5] Because it is compared according to id of concepts, the change of id is regarded as a deletion and an addition of the concept.

2) To refuse the change

2-1) To modify the influenced ontology for compensation of the change; As far as preserving the consistency of the dependency, the developer modifies his/her ontology against the change to cancels the influence of the change.

2-2) To stay compliant with the previous version of the changed ontology; Under controlling versions of the ontologies, the dependency is kept without any modification. After that, when the influencing ontology would be changed so as to be acceptable, the dependent one would adapt to the change and the consistency would be recovered.

2-3) To break the dependency; In order to make the influenced ontology independent of the changed one, reproductions of imported concepts whose change influences on it are redefined as new concepts in the dependent ontology. It implies the dependency on the influencing ontology is broken.

1-1), 1-2) and 2-1) correspond to replacement reproductions of imported concepts with new reproductions based on changed concepts. In 1-1) and 1-2), the developer modifies his/her ontology after the replacement. 2-2) corresponds to do nothing, and 2-3) corresponds to redefinition as discussed above.

We investigated the patterns of the change and the possible way of modification to keep the consistency of the dependency for each pattern. The patterns of the change include the cases where a concept has been deleted, the label has been changed, a slot of a concept has been deleted and so on. For all the cases, we come up with 17 types of change of concepts according to the kind of dependency. And, as the countermeasures for the change, we devised 71 ways of modification. The influenced ontology is modified based on these countermeasures. The details are discussed in our previous work [3]. Though the target of our investigation is a frame language used in Hozo, it can be translated into OWL. Therefore, we suppose most of the patterns are applicable to OWL.

4. Implementation

We have implemented our framework in our environment for building/using ontology: Hozo. Here, we summarize how Hozo supports distributed and cooperative construction of ontologies.

4.1. Overview of Hozo

The features of Hozo include 1) Supporting role representation [4, 5], 2) Visualization of ontologies in a friendly GUI, and 3) Distributed development based on management of dependencies between ontologies. Hozo is composed of Ontology Editor, Onto-Studio (a guide system for ontology design), Ontology Server and Ontology Manager (Fig.4). The ontology editor provides a developer with a graphical interface through which they can browse and modify an ontology locally. The instance models can be developed using Model Editor which is a sub system of the Ontology Editor. The ontology server stores and manages ontologies under access control and version management. Developers can access and browse them through the

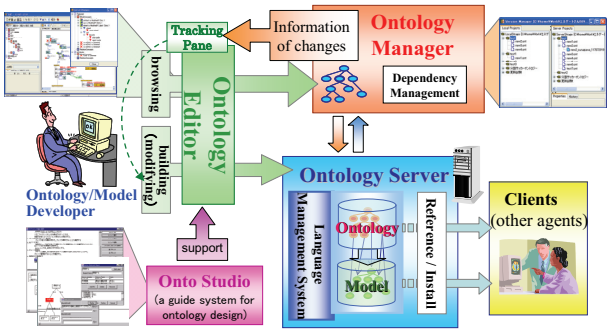


Fig.4. Architecture of Hozo

ontology manager. Furthermore, the ontology editor of Hozo provides a user support module to maintain consistencies of the dependencies among ontologies, called Tracking Pane. Hozo's native language is XML-based frame language and ontologies can be exported in OWL [6], and RDF(S). It also can import OWL partially[6]. The latest version of Hozo is published at the URL: http://www.hozo.jp.

4.2. Version Management and Access Control through Ontology Manager

Hozo can use a general file server as the ontology server. It uses a shared folder on the network or a WebDAV folder to store and share ontologies. The ontologies are managed by changing filenames and storing folders according to their dependencies and versions. The users can share ontologies through a local area network or the Internet. This simple mechanism makes it possible for the users to set up their own ontology server easily without complicated procedures. The user also can switch the ontology server to the other if necessary.

The ontology manager (Fig.5) acts as a bridge between the personal space (in a client) and the shared space which the ontology server provides. It carries out the following functions:

1. To show the latest information on the ontology modules such as "updated", "locked by another developer" and so on.
2. Access control to ontology modules (lock and unlock)
3. Version management of ontology modules
4. To search concepts defined in other ontology modules
5. Synchronize ontology modules in clients with those in the server

4.3. Dependency Management among Ontology modules

When the developer finds reusable concepts defined in other ontologies which are published in the server by other developers, he can import them to his ontology. The

[6] The OWL import mechanism is under improvement.

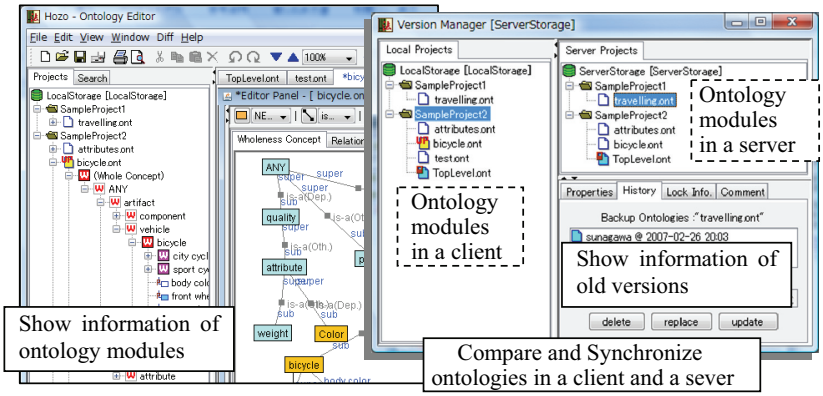


Fig.5. A snapshot of Ontology Manger.

ontology manager supports him to import the concepts through *Import Dialog* of the ontology manager. The dialog shows concepts in the selected ontology by tree structure based on *is-a* relation of them, and the developer selects concepts which he wants to import to his ontology. Then, the system finds all the concepts depended by the selected concepts, forms its dependency relations according to their relations, and finally reproductions of them are imported to his ontology through the procedure discussed in section 3.3. In the ontology editor, reproductions of imported concepts are represented with different color from other concepts, and the developer cannot modify[7] them to keep consistencies of ontologies.

4.4. Analysis of Changes of Depended Ontologies and Their Influences

Ontology Manager shows developers which ontology has been changed. To maintain the consistency of dependency, the developer should get more information on, for example, what concepts/slots in the depended ontology have been changed and which concepts in his ontology are influenced by the changes. Hozo shows such information on the tracking pane and the browsing pane of its ontology editor.

The tracking pane lists the changes in depended ontologies which influence on his ontology (Fig.6). Those changes are classified in three types (deletion, modification and addition), and their types are represented by icons. The changes are shown by nodes with icons in a tree structure, and the developer can know which concepts are influenced by the change through child nodes of the nodes. By clicking a node representing a concept, the selected concept in the ontology is pointed in the browsing pane of ontology editor. In the browsing pane (Fig.7), the ontology is visualized in network structures, and the changed concepts are represented by the same icons[8] as

[7] The developer can use imported concepts to define another concept. For example, he can define sub classes of them.

[8] In the browsing pane, sky blue nodes represent imported concepts from depended ontologies. Therefore, only sky blue nodes can have the icons because the changes appear only on the imported concepts in the distributed ontology development.

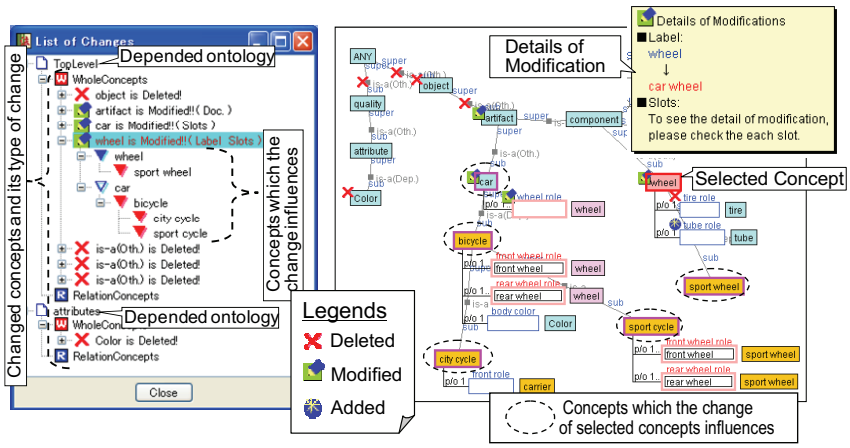


Fig.6. Tracking Pane. Fig.7. Representation of changes on Browsing Pane.

tracking pane shows. When the developer selects a changed concept, the concepts influenced by the change are highlighted in the browsing pane, and then, if the change type of the selected concept is modification, the details are shown.

4.5. Modifying the Ontology to Keep the Consistency

To keep the consistency of the ontology, Hozo suggests possible countermeasures for coping with each of the changes to the developer. These countermeasures are devised through our investigation on conceptual dependencies of ontologies and the change type of imported concepts discussed in section 3.5. In the beginning, Hozo shows developers two major strategies: to accept the change and to reject it. The former corresponds to 1-1), 1-2) and 2-1)[9] discussed in section 3.5. The difference among them depends on the way of modifications after the acceptance of the change. The latter corresponds to 2-3) and implies to redefine the changed concept in his ontology. If the user chooses neither to accept nor to reject the change, it corresponds to 2-2).

For example, if the change type is modification of an imported concept, acceptance of the change corresponds to replacement of the imported concept with the modified one. If the change type is deletion of imported concepts, the acceptance corresponds to deletion of them. Developers can apply these countermeasures by selecting it through a popup menu in the browsing pane. After applying countermeasures, he edits his ontology for coping with the change if necessary. In such a case, it is helpful for him that the system shows the concepts influenced by the change. Furthermore, if he needs advanced strategies, the system shows him all countermeasures[10] with their details in a harmonizing pane.

[9] This strategy means that the user accepts the change and then he/she modifies against the change to cancel the influence of it.

[10] We have not implemented some of advanced countermeasures yet. But, we suppose the two major strategies are enough for coping with the change in a lot of cases.

5. Related Work

Protégé has a semi-automatic tool for ontology merging and alignment named PROMPT [7]. It performs some tasks automatically and guides the user in performing other tasks. PROMPT also detects possible inconsistencies in the ontology, which result from the user's actions, and suggests ways to remedy them. For ontology evolution in collaborative environments [8], Protégé provides two functions: Change-management plugin which stores a list of class-wide changes with annotations and shows history of the change to the user, and Client-Server mode which support synchronous ontology editing by multiple users. SWOOP [9] also supports collaborative annotation for discussing and version control using change logs. But they does not support distributed construction of modularized ontologies discussed in section 2. Their methods for version control are also different form Hozo. They use change logs, but Hozo does not use them and analyzes the changes by comparing ontology with its old version. The approach of Hozo is applicable to ontologies on the Web without their change logs.

DILIGENT [10] and ONKI [11] supports distributed development of ontology through shared space for ontologies in the way as Hozo. But they do not have functions to suggest countermeasures for coping with each of the changes to the developer when depended ontologies are modified. KAON and Hozo focus on that changes in an ontology can cause inconsistencies in other dependent ontologies. And, in order to ensure their consistencies, they propose deriving evolution strategies [12, 13]. But it does not provides strategies which reduce the influences against the changes although Hozo suggests them (e.g. deletion of a concept can be canceled by redefining it in another ontology). The difference is caused by different treatment of relationship between depended ontologies and dependent ontologies.

[14] proposed algorism for modularization of OWL ontology. We have not considered how to modularize ontology. It is one of our future works.

6. Conclusions and Future Work

In this paper, we discussed a framework for distributed and cooperative ontology development. The maintenance of consistencies among modularized ontologies is an essential issue especially in a distributed development. Our framework contributes to resolving the issue based on management of dependencies between ontology modules. The same framework also can support to construct a single ontology by many developers cooperatively. Furthermore, we have implemented the framework in our ontology development environment: Hozo. It supports distributed and cooperative ontology construction by different developers through LAN and Internet. Its functions for distributed ontology construction have been used by some researchers and got favorable comments by them. The latest version of Hozo is open to the public on the website (http://www.hozo.jp).

As future work, the authors plan to enhance our system according to the following future plan: (1) Functions to deal with OWL ontology. For example, we suppose to use OWL properties such as owl:imports and owl:priorVersion for management of

ontology on the Web. (2) Evaluation and reconsideration of strategies for keeping consistencies. (3) Consideration of appropriate modularization. (4) Maintenance of consistency among ontologies and its instance models based on our framework.

Acknowledgments

The authors are grateful to Mr. Mamoru Ohta for his support to implement our system.

References

1. Seidenberg, J., Rector, A.: Web ontology segmentation: Analysis, classification and use. In: 15th International World Wide Web Conference, Edinburgh, Scotland (2006)
2. Noy, N.F., McGuinness D.L.: Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 (2001)
3. Sunagawa, E., Kozaki, K., Kitamura, Y., Mizoguchi, R.: An Environment for Distributed Ontology Development Based on Dependency Management, In: 2nd International Semantic Web Conference, pp. 453--468, Florida, USA (2003)
4. Kozaki K., Kitamura, Y., Mizoguchi, R.: Hozo: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of "Role" and "Relationship", Proc. of EKAW2002, pp.213-218, Siguenza, Spain, 2002.
5. Mizoguchi, R., Sunagawa, E., Kozaki, K., Kitamura, Y.: A Model of Roles in Ontology Development Tool: Hozo. J. Applied Ontology (to appear)
6. Kozaki K., Sunagawa, E., Kozaki, K., Kitamura, Y., Mizoguchi, R.: Role Representation Model Using OWL and SWRL, In: 2nd Workshop on Roles and Relationships in Object Oriented Programming, Multiagent Systems, and Ontologies, Berlin (2007)
7. Noy, N.F., Musen, M.A.: The PROMPT suite: Interactive tools for ontology merging and mapping. International Journal of Human-Computer Studies, 59(6), pp.983—1024 (2003)
8. Noy N., Chugh A., Liu W. and Musen M.: A Framework for Ontology Evolution in Collaborative Environments. In: 5th International Semantic Web Conference, Athens, GA, USA (2006)
9. Kalyanpur, A., Parsia, B., Sirin, B., Cuenca-Grau, B., Hendler, J.:Swoop: A 'Web' Ontology Editing Browser, Journal of Web Semantics Vol 4(2), pp. 144-153 (2005)
10. Tempich, C., Pinto, H.S., Sure, Y., Staab, S.: An Argumentation Ontology for DIstributed, Loosely-controlled and evolvInG Engineering processes of oNTologies (DILIGENT). In: The 2nd European Semantic Web Conference, Greece, pp. 241-256 (2005)
11. Valo, A., Hyvonen, E. Komurainen, V.: A Tool for Collaborative Ontology Development for the Semantic Web, in: Proc. of International Conference on Dublin Core and Metadata Applications 2005, Madrid, Spain (2005)
12. Stojanovic, L., Maedche, A., Motik, B. Stojanovic, N: User-driven Ontology Evolution Management, Proc. of EKAW 2002, Madrid, Spain, pp. 285-300 (2002)
13. Maedche, A., Motik, B., Stojanovic, L., Studer, R., Volz, R. : An Infrastructure for Searching, Reusing and Evolving Distributed Ontologies, The Twelfth International World Wide Web Conference, Budapest, Hungary (2003)
14. Aquin M., Sabou M., and Motta E.: Modularization: a Key for the Dynamic Selection of Relevant Knowledge Components, The First Workshop on Modular Ontologies (2006)

Vocabulary Patterns in Free-for-all Collaborative Indexing Systems

Wolfgang Maass, Tobias Kowatsch, and Timo Münster

Hochschule Furtwangen University (HFU)
Robert-Gerwig-Platz 1, D-78120 Furtwangen, Germany
{wolfgang.maass,tobias.kowatsch,timo.muenster}@hs-furtwangen.de

Abstract. In collaborative indexing systems users generate a big amount of metadata by labelling web-based content. These labels are known as tags and form a shared vocabulary. In order to understand the characteristics of that vocabulary, we study structural patterns of these tags by implying the theory of self-organizing systems. Therefore, we utilize the graph theoretic notion to model the network of tags and their valued connections, which represent frequency rates of co-occurring tags. Empirical data is provided by the free-for-all collaborative indexing systems Delicious, Connotea and CiteULike. First, we measure the frequency distribution of co-occurring tags. Secondly, we correlate these tags towards their rank over time. Results indicate a strong relationship among a few tags as well as a notable persistence of these tags over time. Therefore, we make the educated guess that the observed collaborative indexing systems are self-organizing systems towards a shared vocabulary building. Implications on the results are the presence of semantic domains based on high frequency rates of co-occurring tags, which reflect topics of interest among the user community. When observing those semantic domains over time, that information can be used to provide a historical or trend-setting development of the community's interests, thus enhancing collaborative indexing systems in general as well as providing a new tool to develop community-based products and services at the same time.

Key words: Metadata, tagging, shared vocabulary, online community, collaborative software, self-organizing system

1 Introduction

Cooperative, distributed labelling of content in the worldwide web is called collaborative indexing or social tagging. Within a collaborative indexing system users annotate different contents e.g.: events[1], video clips[2], music[3], pictures[4],

[1] http://upcoming.org
[2] http://youtube.com
[3] http://last.fm
[4] http://flickr.com, http://espgame.org

articles and references[5], weblogs[6] or websites[7]. These collaborative indexing systems facilitate mass categorization establishing so-called folksonomies, which is a bottom up categorization made by a large user base.

A collaborative indexing system has basically two features. First, it is used for future retrieval of self-indexed content. Secondly, it provides recommendations, which are based upon the co-occurrence of highly used tags within all annotations, whereas we call one single process of annotation an indexing task.[8] The recommendations are shown to the user by committing a tag query. For instance: content tagged with *html* will be frequently tagged with *css* as well.

The data collected within an indexing task contains the name of the user, an url linking to the content, one or more tags and time-stamp information. Therefore, the data within a collaborative indexing system is basically a network of users, tags and content in a given period of time. All tags together represent the shared vocabulary of the user community. In this paper we study the structural patterns of that vocabulary, thus focusing only on the partial network of tags. Analyzing this partial network requires some constructs of the graph theory. We assume the shared vocabulary to be a self-organizing system by means of the systems theory [1]. Hence, stable patterns as well as specific correlations are determined throughout the vocabulary.

In addition, implications on these patterns are presented. To support the requirements of self-organizing systems by reducing external restrictions and forces we choose the free-for-all collaborative indexing systems Delicious, Connotea and CiteULike for empirical data extraction, where any user can index any content element. Thus, indexing rights are not restricted as identified by Marlow et al. [2].

This paper starts with related work covering collaborative indexing systems and the systems theory. Then, we hypothesize two assumptions regarding stable patterns within the vocabulary. Afterwards, we build up a model based on the graph theoretic notion, clarify the methodic approach and present the empirical data used to prove the assumptions. Subsequently, we present and discuss the results of our analysis and draw implications on them. Finally, we give an outlook on further research.

2 Related Work

A general review on collaborative indexing systems is given by Voss [3]. Mathes [4] discusses the organization of information via tags and points out that user generated metadata is of an uncontrolled nature and fundamentally chaotic compared to a controlled vocabulary. But he also mentions that collaborative index-

[5] `http://citeulike.org`, `http://connotea.org`, `http://bibsonomy.org`

[6] `http://technorati.com`

[7] `http://del.icio.us`, `http://myweb.yahoo.com`

[8] There may also exist other recommender implementations, but we focus on the co-occurrence of highly used tags because this information is freely accessible on the web.

ing systems are highly responsive to the users needs and their vocabulary by involving them into the process of organization. Vander Wal [5] distinguishes between broad and narrow folksonomies depending on the amount of users, who tag one specific content element. He also defines the difference between pure tagging and folksonomy tagging.

Voss [6] discovers power law distributions of tag frequency rates in Delicious and Wikipedia supporting the presence of self-organizing systems. Hotho et al. [7] and Quintarelli [8] find power law distributions according to collaborative indexing systems, too. Lund et al. [9] measure a power law distribution of user shared tags within Connotea. Results of Golder and Huberman [10] show regularities of dynamic structures within Delicious. Moreover, they introduce a classification on the semantics of tags as well as Zhichen et al. [11].

Wu et al. [12] distinguish the potential of collaborative indexing systems as a technological infrastructure for acquiring social knowledge. Millen et al. [13] study the deployment of a collaborative indexing system within a company and highlight the remarkable acceptance rate of the users as well as its personal and organizational usefulness. In addition, Damianos et al. [14], Farrell and Lau [15] as well as John and Seligmann [16] also examine the potential of collaborative indexing systems for the enterprise covering people's expertise, social networks and the integration of those systems in existing collaborative applications.

An early classification of collaborative indexing systems is done by Hammond et al. [17] confronting scholarly and general resources with links and web pages. In a more detailed classification Marlow et al. [2] distinguish the design of a system and present several user incentives. Heymann and Garcia-Molina [18] develop an algorithm, which generates a hierarchical taxonomy of a tag network. For the same purpose Mika [19] uses social network analysis on the network of users, tags and content. Hotho et al. [7] develop a search algorithm for folksonomies to find communities of interest within collaborative indexing systems. Cattuto et al. [20] design a stochastic model for the analysis of indexing tasks over time consisting of tags and users. Dubinko et al. [21] visualize tags over time with data from Flickr, whereas Zhichen et al. [11] propose an algorithm for tag suggestions to support the user within an indexing task. An overview of self-organizing systems is given by Heylighen [1].

3 Motivation

As mentioned above, this paper deals with the partial network of tags. The concept of tags is central in collaborative indexing systems. The same tags used by different users to annotate similar content show a common understanding of the users. The set of all tags utilized by the user community represents the shared vocabulary. Users and content elements are linked to each other through tags, which are also directly connected when they are used together within one indexing task. Figures 1 and 2 are representing such an indexing task as well as the resulting network of the tags *sports*, *worldcup* and *soccer*. Due to the current work, the value of those tag connections is an essential dimension, which

is based on the frequency rate of tags co-occurring within all indexing tasks. A prerequisite for a measurement of this frequency is the bag-model for aggregation of tags, in which multiple tags can be assigned to one resource by multiple users as discussed by Marlow et al. [2].

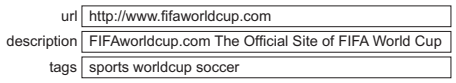


Fig. 1. Graphical input mask for an indexing task

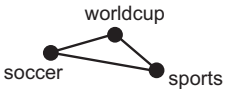


Fig. 2. Resulting network of the indexing task in Fig. 1

Prior work on stable patterns suggests that collaborative indexing systems are self-organizing systems [10, 2, 6, 8, 9]. The vocabulary - consisting of tags and generated within all indexing tasks by all users - is a part of this system, which organizes its structure by itself, without a centralized control mechanism. The users of a collaborative indexing system generate this vocabulary in a decentralized approach, not even aware of it. On its own this system evolves over time into a more stable state.

Contrary to the aforementioned work, we explore patterns emerging out of co-occurring tags. Therefore, we want to know if the power law distribution, which is common in broad folksonomies [7, 9, 8], is also applicable to the structure of co-occurring tags. This would represent a community's vocabulary, which consists of a few tags co-occurring with high frequency rates and many tags co-occurring with low frequency rates. Such a pattern - we call it tag economics - would indicate a strong consensus on a particular subpart of the community's vocabulary, from which particular interests of the users can be identified. Due to these considerations, we hypothesize the relation of co-occurring tags as follows:

H1 Let T_i be a tag and T_i^j all tags co-occurring with T_i. Then the ranked frequency distribution of all valued connections from T_i to T_i^j follows a power law curve.

Additionally, we focus on the frequency dynamics of tags over time depending on their position in the aforementioned frequency distribution. We assume that tags co-occurring with high frequency rates (higher position on the power law curve) are more stable over time than tags co-occurring with low frequency rates. This would represent persistence of the community's interests or, when tags with high frequency rates change to a low position, one can suggest a shift of the community's common understanding. Therefore, the current work has the second objective to examine the relationship of the frequency rates of co-occurring tags over time. We hypothesize this relationship as follows:

H2 The higher the frequency rates of the tags T_i^j, the more stable are they over time.

4 Model

Let an indexing task be a quadruple comprised of $< user, url, timestamp, tag^* >$. One user enters an url with none, one or more tags into a collaborative indexing system at a certain time. Only two entities are important according to our hypotheses, namely *timestamp* and *tag*. Therefore, The community's vocabulary is modelled as an undirected, valued and finite graph V within a given period of time δ. This period of time is essential, because the frequency of time based indexing tasks is subject to fluctuations, which occur in the course of a day, a week or month. Furthermore, δ can be used to affect directly the size of the vocabulary V to ease the analysis.

The vocabulary V consists of a set of nodes (here tags) and a set of valued links, which represent the frequency values of co-occurring tags. Hence, we refer to this vocabulary as the network of tags, too. The links are undirected since each tag i, which co-occurs with a tag j, also means that the tag j co-occurs with the tag i, respectively. To better handle these frequency values, the vocabulary can be described by a symmetric frequency matrix F, such that the value on the ith row and jth column represents the frequency rate of the co-occurring tags i and j over all indexing tasks within δ, denoted as $f(i, j)$. Self references are excluded since we focus only on co-occurring tags. Thus, the diagonal values $f(i, j)$ with $i = j$ are always zero. Figure 3 exemplifies an undirected, valued graph of the vocabulary V, whereas Fig. 4 shows the corresponding frequency matrix F. Based upon this graph theoretic notion and the corresponding frequency matrix, we are able to illustrate and compute the frequency distribution of co-occurring tags.

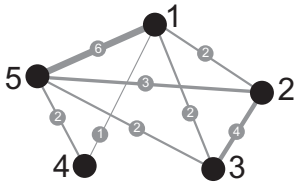


Fig. 3. Undirected, valued graph of the vocabulary V including 5 tags

i \ j	1	2	3	4	5
1	0	2	2	1	6
2	2	0	4	0	3
3	2	4	0	0	2
4	1	0	0	0	2
5	6	3	2	2	0

Fig. 4. Corresponding frequency matrix F of the vocabulary V

4.1 Method

A frequency matrix $F(\delta_1)$ is built within a given period of time. Afterwards, the frequency values $f(i, j)$ for each tag T_i are summed up. Consecutively, those

cumulative frequency rates are ranked by size and confined by a limit L. This approach eliminates tags T_i with low cumulative frequency values of co-occurring tags T_i^j, because they cannot contribute any meaningful values for co-occurring tags and are therefore not relevant for further calculations. Then, N tags T_i with maximum $N_{\max}$, medium N_{med} and minimum $N_{\min}$ cumulative frequency rates are identified. Afterwards, the frequency distribution of all tags T_i^j co-occurring with each tag T_i is calculated from this selection and subsequently ranked by size. Finally, the values of these frequency distributions are normalized and utilized to conduct a curve estimation regression statistic based on the power model, whose equation is $f(r) = \beta_0 r^{\beta_1}$ with $f(r)$ estimating the frequency rate depending on the frequency rank r of a tag T_i^j. The results of the regression statistics are used to prove hypothesis 1.

The aforementioned N tags T_i are also used to prove the second hypothesis. Hence, N tags T_i with maximum $N_{\max}$, medium N_{med}, and minimum $N_{\min}$ cumulative frequency rates are identified. Afterwards, the frequency rates for each pair of co-occurring tags are written down in a time series each lasting δ_2 over I iterations. To measure the stability between co-occurring tags T_i and T_i^j, the difference D from the mean frequency of each tag co-occurrence in $N_{\max}$, N_{med}, and $N_{\min}$ is calculated over all I iterations, so they can be compared afterwards.

4.2 Empirical Data

The empirical data for the analysis was extracted from the collaborative indexing systems Delicious, Connotea and CiteULike. This information is freely accessible. Indexing rights are based on a free-for-all principle [2], thus supporting the requirements of self-organizing systems by reducing external restrictions and forces. The content is respectively of textual nature. The selected collaborative indexing systems differ in the community's size and the quantity of indexing tasks, the amount of tags, as well as the period of time in which the data was gathered. Furthermore, all indexing systems use a bag-model to aggregate tags, which is essential for our approach as mentioned in Sect. 3. Table 1 provides detailed information about the empirical data.

Table 1. Empirical data

Indexing system	Delicious	Connotea	CiteULike
Period of measurement	09/01/06 – 10/01/06	01/01/06 – 10/01/06	09/17/06 – 10/01/06
Indexing tasks (It)	452 806	92 333	3 798
It incl. at least 2 tags	269 737 (60%)	56 289 (61%)	2 430 (25%)
Tags incl. doublets	1 169 396	250 293	9 765
Distinct tags	130 776 (11%)	41 707 (17%)	3 659 (37%)
Distinct users	121 197	3 929	633
Users with at least 2 It	70 519 (58%)	2 722 (69%)	408 (64%)

5 Results

5.1 Hypothesis 1: Power Law Distribution of Co-occurring Tags

The ranked frequency distribution $f(r)$ of T_i^j tags co-occurring with a tag T_i is illustrated in Fig. 5. Thus, a power law distribution is clearly apparent in the shared vocabulary, as to be expected from a broad folksonomy like Delicious. There are many tags T_i^j with low frequency rates of co-occurring tags and few with very high frequency rates. This result is proved by the cumulative discrete co-occurrence distribution in Fig. 6, which illustrates the discrete frequency distribution. There is a remarkable gap between tags, which co-occur with only one single tag, and tags co-occurring with multiple other tags. Towards the high co-occurrence rates the curve decreases rapidly as the logarithmic scale demonstrates. Frequency rates of co-occurring tags above 100 lead to absolute frequency rates less then ten. Similar results are provided by the collaborative indexing systems Connotea and CiteULike.

The visual observations in Fig. 5 and 6 can be confirmed statistically. Therefore, Table 2 shows the median of squared reliability ($\bar{R}^2$), the median degree of freedom ($\bar{F}$) as well as the exponent $\bar{\beta}_1$ according to the power law curve estimation algorithm[9] over all corresponding tags within $N_{\max}$, N_{med}, and $N_{\min}$. Compared with other curve estimation algorithms, the power model performed best by far.

In particular, data from Delicious with 2007 co-occurring tags T_i^j as a median degree of freedom shows a remarkable reliability of .96 by only .01 standard deviation for $N_{\max}$. Values with less reliability values lie nearby .80, which is still acceptable although standard deviation values show higher dynamics. Additionally, a decrease of the exponent $\bar{\beta}_1$ can be observed related to the degree of freedom by considering the data of Delicious and Connotea. This can be referred to a smoother power law curve, when less tags co-occur with a tag T_i. For this reason, the relative low degree of freedom according to CiteULike can be neglected to identify the aforementioned effect.

Due to these facts, the first hypothesis is supported by the empirical data. It is quite evident that a power law curve of co-occurring tags is obvious for tags T_i with high frequency rates, whereas the co-occurrences of middle and low ranked tags T_i show more dynamics.

5.2 Hypothesis 2: Relation between Rank and Persistence of Tags over Time

Figure 7 shows the dynamics over time of co-occurring tags against their deviations from the mean frequency. As illustrated, co-occurring tags with high frequency rates - values from $N_{\max}^{T_{1-250}}$ - are more stable and have a lower scatter respectively than co-occurring tags from N_{med} or $N_{\min}$. Figure 8 shows the

[9] Statistical software used for curve estimation: SPSS, Version 15.0.1, SPSS Inc. Chicago, USA

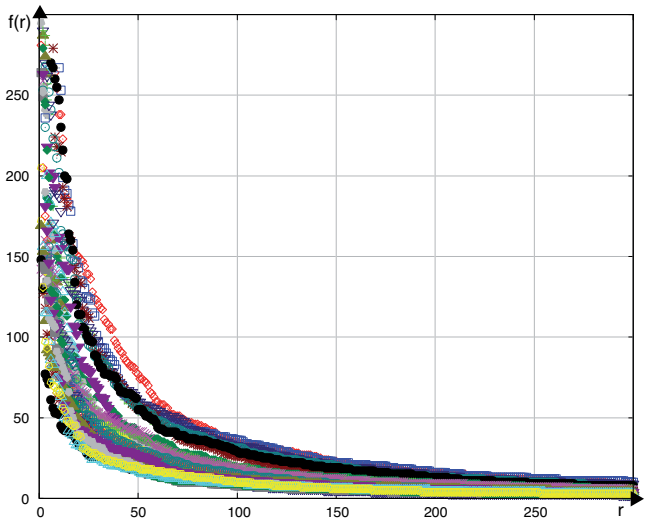


Fig. 5. Extract from the ranked frequency rates $f(r)$ of T_i^j tags co-occurring with 30 tags T_i (different symbols) based on Delicious, δ_1: 09/10/06 – 09/19/06

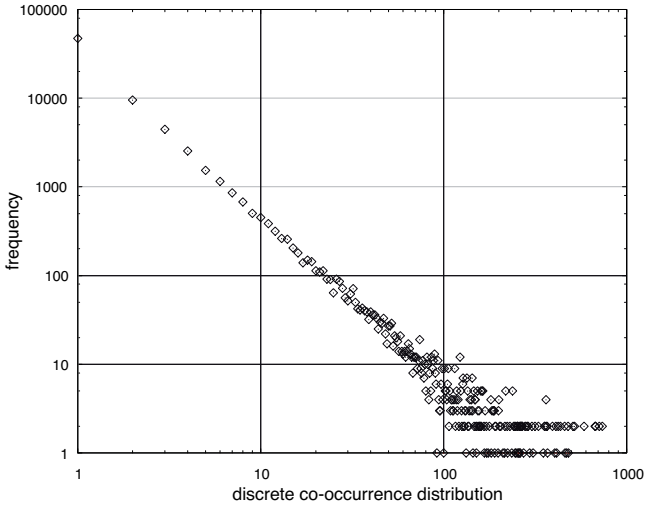


Fig. 6. Discrete co-occurrence distribution of tags T_i and T_i^j from Fig. 5 and the corresponding frequency values

Table 2. Curve estimation regression statistics based on the power law $f(r) = \beta_0 r^{\beta_1}$, $\bar{R}^2$: median of squared reliability, $\bar{F}$: median degree of freedom, standard deviations are provided in brackets

Indexing system	Delicious	Connotea	CiteULike
N / L	30 / 25	30 / 25	20 / 10
δ_1	09/10/06 – 09/19/06	01/01/06 – 09/25/06	09/15/06 – 09/25/06
$\bar{R}^2$ / $\bar{F}$ / $\bar{\beta}_1$			
for $N_{\max}$	.96 (.01) / 2007 / -.96 (.08)	.93 (.10) / 638 / -.82 (.28)	.81 (.10) / 58 / -.46 (.18)
for N_{med}	.82 (.08) / 152 / -.51 (.12)	.86 (.06) / 88 / -.58 (.29)	.79 (.12) / 32 / -.47 (.32)
for $N_{\min}$	.78 (.11) / 73 / -.42 (.21)	.84 (.11) / 55 / -.53 (.28)	.82 (.20) / 20 / -.65 (.40)

relative frequencies of two co-occurring tags from $N_{\max}$ and N_{med} in contrast to each other over 30 iterations with $\delta_2 = 1\,\text{day}$. This figure illustrates that the interval of N_{med} shows higher variations than the interval of $N_{\max}$.

The basic data from all examined collaborative indexing systems with the average deviation of the mean frequency values $\bar{D}$ over $N_{\max}$, N_{med}, and $N_{\min}$ is shown in Table 3. As a result, δ_2 and the number of indexing tasks within δ_2 are affecting the dynamics. Those in Connotea and CiteULike are much lower than the dynamics in Delicious. This often causes co-occurring tags to appear only in a small degree over all iterations and therefore, they stabilize only on a low level with low frequency rates. Thus, a small deviation from the average frequency rate over all iterations is not always indicating a high position of co-occurring tags on the power law curve. There are also situations, where tags stabilize on low frequency rates. The stability alone is therefore not a sufficient criterion for the occurrence of high frequencies. In fact, the absolute frequency rates must be observed for a positioning on the power law curve besides the deviation.

As a result, correlations between high frequency rates and their persistence over time can be concluded, but not vice versa. A change of that persistence is therefore only significant for a shared vocabulary, if the deviation of the average value appears on high frequency rates. Nevertheless, the second hypothesis is also supported by the figures of Table 3, although with less explanatory power compared to the findings of hypothesis 1.

Table 3. Deviation over the average frequency

Indexing system	Delicious	Connotea	CiteULike
N / L	250 / 30	30 / 10	15 / 5
δ_2	1 day,	1 month,	1 day,
	09/01/06 – 09/30/06	01/01/06 – 08/31/06	09/16/06 – 09/30/06
$\bar{D}(N_{\max})$	14.1%	22.2%	15.1%
$\bar{D}(N_{\text{med}})$	19.4%	24.0%	16.1%
$\bar{D}(N_{\min})$	19.8%	27.0%	16.7%

5.3 Implications

As shown in section 5.1, we observe a frequency distribution of co-occurring tags which follows a power law curve. The examined collaborative indexing systems

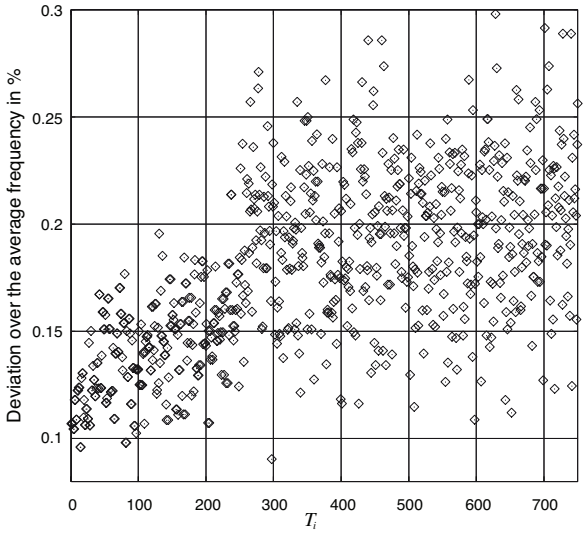


Fig. 7. Deviation over the average frequency in % of 750 tags T_i and all related tags T_i^j for $N_{\max}^{T_{1-250}}$, $N_{\text{med}}^{T_{251-500}}$ and $N_{\min}^{T_{501-750}}$ based on Delicious, between 09/01/06 – 09/30/06

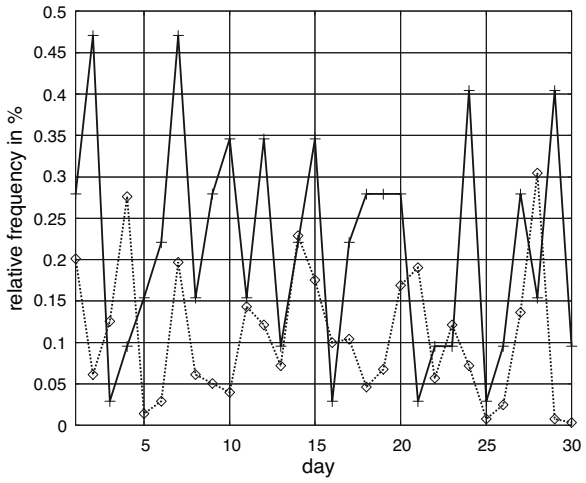


Fig. 8. Relative frequency comparison in % of 1 high (dashed) and low positioned tag T_i from Fig. 5 with $\delta_2 = 1\,\text{day}$ and 30 iterations based on Delicious, between 09/01/06 – 09/30/06

support a distributed approach without central control mechanisms, favoring self-organization. The observed distribution of tags means that the users have a strong consensus at least on a particular subpart of the shared vocabulary, since co-occurring tags with high frequency rates build a semantic domain. This shows some sort of tag economics within a collaborative indexing system.

A further aspect indicating a self-organizing system is the resilience of the system. Accidental errors, e.g., typos or willful sabotages of the system by users have negligible effects, because single users cannot tip the scales of a power law curve. In addition, the construct of indexing support through pre-defined tags, which is suggested to consolidate the tag usage [11, 2], would additionally support these findings by diminishing the limits of the uncontrolled vocabulary such as polysemy, synonymy/uniformity and basic level variation problems [10, 22]. Hence, we suggest higher frequency rates within the top ranked tags as well as lower rates within low ranked ones as fundamental impacts of the indexing support construct.

Another feature of a self-organizing system is the adaptation of environmental changes. In terms of a collaborative indexing system, these changes can be referred to as a shift of the community's interest, which is likewise reflected in a structural change of the vocabulary. Hence, semantic domains based upon co-occurring tags with high frequency rates may change. For instance, if the position of a tag T_i^j alters over time by means of an increase or decrease of the frequency rates according to T_i, then this progress suggests a structural change within the vocabulary and vice versa. The higher the position of this tag on the power law curve, the more significant is the structural change of the vocabulary. When this dynamic information is monitored one can observe a historical or trend-setting development of the vocabulary based upon the time-stamp of selected indexing tasks. Those trend curves of the vocabulary suggest changes within the community's interest and are useful for the particular user when searching for content elements, users or tags in the time domain.

6 Conclusion and Future Work

In this paper we studied structural patterns of user generated vocabularies within the free-for-all collaborative indexing systems Delicious, Connotea and CiteULike. The theory of self-organizing systems was implied to hypothesize patterns within those vocabularies. We built up a model based on the graph theoretic notion consisting of tags and their valued connections. This was required to calculate the frequency distribution of tags that co-occur with others, as well as to correlate those tags towards their frequency rate over time.

Results indicate that only a few co-occurring tags exist with high and many with low frequency rates, thus following a power law curve. In addition to that, co-occurring tags with high frequency rates proved to be more stable over time than those with low rates. The results were also depending on the quantity of indexing tasks. For instance, the measured values of CiteULike yielded less explanatory power than the values of Delicious. Implications are drawn through

the presence of semantic domains, which are based on co-occurring tags with high frequency rates and the shift of common interests among the user community, if those high rates are fundamentally changing over time. The resulting information can be used to provide a historical or trend-setting development of the vocabulary and would not only be useful for the particular user but would also support enterprises to develop products and services, which may depend on or at least involve the interests and trends of online communities.

Due to the current work, the development of algorithms for trend information and historical time series based on the frequency distribution of co-occurring tags is an interesting area for further research. A common understanding of the user community is expressed through the tag network comprised of valued links with high frequency rates. In addition, semantic domains of more than two co-occurring tags can also be identified with techniques of the social network analysis such as centrality measurements or clustering. This network alters dynamically in a self-organizing way over time suggesting new topics or events of social, academic, technical or economic nature. Defining triggers on the observed power law curve to identify those variances requires further clarification, but would be very useful by supporting users, enterprises or public organisations in upcoming decisions.

Moreover, there is still a challenge in collaborative indexing systems featuring low indexing rates within a given period of time. This applies especially for those systems deployed in companies. Therefore, it is essential to find techniques, which permit major vocabulary coherence in such minimal systems and boost the significance of the common understanding.

References

1. Heylighen, F.: The science of self-organization and adaptivity. In: The Encyclopedia of Life Support Systems, Oxford, UK, Eolss Publishers (1999)
2. Marlow, C., Naaman, M., Boyd, D., Davis, M.: Ht06, tagging paper, taxonomy, flickr, academic article, to read. In: HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and Hypermedia, New York, ACM Press (2006) 31–40
3. Voss, J.: Tagging, folksonomy & co - renaissance of manual indexing? ArXiv Computer Science e-prints (January 2007)
4. Mathes, A.: Folksonomies - cooperative classification and communication through shared metadata. Technical report, Graduate School of Library and Information Science, University of Illinois (December 2004)
5. Vander Wal, T.: Explaining and showing broad and narrow folksonomies `http://www.personalinfocloud.com/2005/02/explaining_and_.html` (February 2005)
6. Voss, J.: Collaborative thesaurus tagging the wikipedia way. ArXiv Computer Science e-prints (April 2006)
7. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Information retrieval in folksonomies: Search and ranking. In Sure, Y., Domingue, J., eds.: The Semantic Web: Research and Applications. Volume 4011 of LNAI., Heidelberg, Springer (June 2006) 411–426

8. Quintarelli, E.: Folksonomies: power to the people. Incontro ISKO Italia - UniMIB Meeting (June 2005)
9. Lund, B., Hammond, T., Flack, M., Hannay, T.: Social bookmarking tools (ii) a case study - connotea. D-Lib Magazine **11**(4) (April 2005)
10. Golder, S.A., Huberman, B.A.: Usage patterns of collaborative tagging systems. Journal of Information Science **32**(2) (April 2006) 198–208
11. Zhichen, X., Yun, F., Jianchang, M., Difu, S.: Towards the semantic web: Collaborative tag suggestions. In: Collaborative Web Tagging Workshop, WWW 2006, 15th International World Wide Web Conference, Edinburgh, IW3C2 (May 2006)
12. Wu, H., Zubair, M., Maly, K.: Harvesting social knowledge from folksonomies. In: HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and Hypermedia, New York, ACM Press (2006) 111–114
13. Millen, D.R., Feinberg, J., Kerr, B.: Dogear: Social bookmarking in the enterprise. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, ACM Press (2006) 111–120
14. Damianos, L., Griffith, J., Cuomo, D.: Onomi: Social bookmarking on a corporate intranet. In: Collaborative Web Tagging Workshop, WWW 2006, 15th International World Wide Web Conference, Edinburgh, IW3C2 (May 2006)
15. Farrell, S., Lau, T.: Fringe contacts: People-tagging for the enterprise. In: Collaborative Web Tagging Workshop, WWW 2006, 15th International World Wide Web Conference, Edinburgh, IW3C2 (May 2006)
16. John, A., Seligmann, D.: Collaborative tagging and expertise in the enterprise. In: Collaborative Web Tagging Workshop, WWW 2006, 15th International World Wide Web Conference, Edinburgh, IW3C2 (May 2006)
17. Hammond, T., Hannay, T., Lund, B., Scott, J.: Social bookmarking tools (i): A general review. D-Lib Magazine **11**(4) (April 2005)
18. Heymann, P., Garcia-Molina, H.: Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical report, Computer Science Department, Stanford University (April 2006)
19. Mika, P.: Ontologies are us: A unified model of social networks and semantics. In: 4th International Semantic Web Conference (ISWC 2005). (2005)
20. Cattuto, C., Loreto, V., Pietronero, L.: Collaborative tagging and semiotic dynamics. ArXiv Computer Science e-prints (May 2006)
21. Dubinko, M., Kumar, R., Magnani, J., Novak, J., Raghavan, P., Tomkins, A.: Visualizing tags over time. In: WWW '06: Proceedings of the 15th international conference on World Wide Web, New York, ACM Press (2006) 193–202
22. Furnas, G.W., Landauer, T.K., Gomez, L.M., Dumais, S.T.: The vocabulary problem in human-system communication. Communications of the ACM **30**(11) (1987) 964–971

Ontology Revision as Non-Prioritized Belief Revision

Mauro Mazzieri and Aldo Franco Dragoni

Department of Electronics, Artificial Intelligence and Telecommunications,
Università Politecnica delle Marche, Ancona, Italy
{m.mazzieri,a.f.dragoni}@univpm.it

Abstract. Ontology revision is the process of managing an ontology when a new axiom or fact would render it inconsistent. So far, the AGM approach to belief revision has been adapted to work with ontologies. However, when multiple sources are contributing uncertain knowledge about a static domain, an approach that doesn't give priority to incoming information and allows to recover previously discarded axioms is more suited.
We describe an ontology revision framework that links symbolic and numerical techniques to allow the consistent evolution of an ontology from the contributions of multiple potentially unreliable sources.

Key words: Ontology revision, belief revision, OWL.

1 Introduction

An explicit specification of a conceptualization for a shared domain of discourse is called an ontology[1]. Hence, changes in ontologies are caused by changes either in the domain, or in the conceptualization, or in the defined specification[2].

Ontology evolution[3] is the process of modifying an ontology in response to a change in the domain (first kind of change) or its conceptualization (second kind). The case of change in the domain is analogous to *belief updating*, thus it can be defined as *ontology updating* (more on this on §3). This work deals with changes in the shared conceptualization: a problem analogous to *belief revision*, thus the name *ontology revision*. The third kind of change refers to a change in the way the conceptualization is formally recorded; this type of change is dealt with in the field of *ontology translation*[4].

Current work on ontology evolution is based on the idea of bringing the AGM belief change theory[5, 6] to work within ontology evolution; Flouris' PhD thesis[3] contains both novel contributions and a survey of the field; [7] depicts the state of the art in AGM-based ontology revision.

However, AGM belief revision is not apt to all kind of ontology changes. One of its principles states that incoming information has a priority: it must belong to the new set of beliefs. This principle works well when the new information represent a certain fact: either a realization of the new contingent state of the world, or a correction of a previous error in conceptualization, or a required

property of the formalization. The principle can not be accepted when the new information represents a new evidence about the world, supposed to be a fixed static entity, while its description is only partial and uncertain. In particular, it can not be accepted in a distributed environment, where multiple potentially unreliable information sources are present. Not only an information from an external source can not be unconditionally accepted (can you trust everything you hear?); also, there is not always a relation between the arrival order of information and their acceptability.

There are many different possibility to discard the principle of priority to incoming information. Hansson[8] makes a survey of different varieties of non-prioritized belief revision, i.e. belief revision in which the new information has no special priority due to its novelty. The problem is when and how to choose if the new information must be accepted. We will follow an integrated approach, already successfully applied to a juridical domain[9], which deals with old and new information as they were come at the same time. This approach relies both on symbolic and numerical techniques and make use of a new principle, called *principle of recoverability*[10] [1]:

> Any previously held piece of knowledge should belong to the current knowledge space if consistent with it.

To circumscribe the work, we will refer to a specific use case. A team of loosely-coordinated domain experts has the duty to build an ontology for their domain. Each team member contributes to the activity building his conceptualization with an editor. The domain is assumed as a fixed static entity, while the conceptualization is constantly changing during the building and refinement process. The work of each member is shared with the other experts in a peer-to-peer way: each member receives the contribution of the other experts. A supporting software must be able to use an ontology revision mechanism to maintain a consistent local ontology to be visualized and used as the basis for further editing. An example of a work session will be shown in section 5.

In the following we will first summarize in an informal way the syntax and semantic of the OWL ontology language (§2). Then, after an introdution to the problem of ontology revision (§3), we will show the proposed revision procedure (§4), both in its symbolical (§4.1) and numerical (§4.2) steps. Finally in §6 we sketch the future research perspectives.

2 Ontology

The OWL web ontology language[12] is the language used for publishing and sharing ontologies on the World Wide Web. OWL is developed as an extension of the RDF[13] knowledge representation language. The language has two specific subsets: OWL DL and OWL Lite. The complete language is called OWL Full

[1] Introduced as the *store and recover principle*[11] and also known as the *principle of persistence*[9].

to distinguish it from the subsets. The DL in OWL DL stands for "Description Logic"[14], a decidable subset of first order logic used for expressing structured knowledge. OWL DL and OWL Lite are both based on description logic; the former is more expressive, while the latter has better computational properties.

In order to introduce the problem of ontology revision and to make the work self-contained, we will give here an informal definition of an ontology language syntax and semantics, roughly correspondant to OWL Lite. The full formal semantics and syntact can be found in [15].

2.1 Syntax

The basic building blocks of an ontology are *classes*, *individuals* and *properties*. A class is related to a set of individuals, called class extension. Properties can be either data-valued, relating individuals to values, or individual-valued, relating individuals to other individuals.

An *ontology* is a set of class axioms, property axioms and facts.

There are two kinds of *class axioms*. A class can be defined as either exactly equivalent to the conjunction of a set of superclasses, or as a subclass of the conjunction of a set of superclasses. A superclass can be either another class, or an anonymous class specified giving constraints on properties.

The allowed restrictions on property values are:

- all the values must be instances of a class (or from a datatype, in the case of data-valued properties);
- some of the values must be instances of a class (or from a datatype, in the case of data-valued properties);
- the cardinality must be at least (or at most, or equal to) either 0 or 1.

Property axioms are used to define properties. A property can be given a super-property, allowing the construction of a property hierarchy. Properties can also be given domains and ranges.

Data-valued properties can be specified as partial functional, i.e. with at most a value. Individual-valued properties can be specified to be functional, inverse-functional, symmetric, transitive, or the inverse of another property.

Finally, a *fact* states that an individual belongs to a class or that an individual's property has a certain value.

2.2 Semantics

An OWL *interpretations* defines:

- a class as a collection of individuals,
- a datatype as a set of literal values,
- a data-valued property as a relation from individuals to literal values,
- an individual-valued property as a relation from individuals to other individuals.

An interpretation I *satisfies* an ontology O if it obeys to all restrictions given by O's axioms and facts.

An ontology O is *consistent* if there is at least an interpretation I which satisfies the ontology.

An ontology O *entails* an ontology O' if each interpretation I which satisfies O also satisfies O'.

3 Ontology Revision

3.1 Belief Revision

Ontology revision has many similarities with belief revision.

Belief revision is the process of rearranging a knowledge base to preserve global consistency while accommodating incoming information. In the AGM theory[5, 6], the belief is formalized as a set of logical statements, (the *belief set*), i.e. a logic theory K described in a formal language L. The belief set is closed under logical consequences. A finite subset B of K such that $K = \mathrm{Th}(B)$ is a *knowledge base* for K. The problem of revision arises when we get a new formula p that makes the knowledge base *inconsistent*. Then, we have to *revise* the knowledge base, retracting some of the beliefs, in order to restore consistency. The revised theory is K^*p. The AGM theory gives three rationality principles affecting K^*p:

Consistency The revised belief K^*p must be consistent.

Minimal change The revision process should alter as little as possible the current belief set.

Priority to incoming information The new information p must belong to the new belief set K^*p.

From these principles eight postulates follow. However, neither the rationality principles nor the postulates univocally define revision.

3.2 Definition of Ontology Revision

Ontology revision is defined as a change of components in ontology[16]. Coeherently with belief revision theory, we define ontology revision as the process of rearranging an ontology to preserve consistency while accomodating changes. Foo[17] presents a summary of issues concening ontology revision from artificial intelligence, philosophy and recursion theory.

Our approach to ontology revision will be based on *belief bases*, a set of sentences not closed under logical consequence, from which a belief set can be derived[18]. Our belief base is an ontology, i.e. a set of axioms and facts. The incoming information is represented as an axiom or a fact, i.e. a TBox or a ABox statement[2]. The problem of revision arises when the new axiom or fact would render the ontology inconsistent.

[2] Another approach, such the one in [19], considers only inconsistences due to objects introduced in the ABox.

The choice to represent changes at the level of single axioms is very fine-grained, but it doesn't forbid to define more complex, higher-level changes[20]. A finer-grade approach, involving the single constraints in class and property axioms, would be problematic as not all combinations are allowed. For example, in OWL Lite, not all properties can have cardinality restrictions placed on them or be specified as functional or inverse-functional[15]. An example of this approach, involving the weakening of the original ontology to accomodate the incoming axiom, is presented in [3].

To choose an ontology revision procedure we have first to understand why an axiom or fact, potentially incompatible with the current ontology, can arrive. We want to point out two different scenarios, demanding a different approach:

- The ontology represents the current state of an evolving world, and the new information reflects a change in the world. The consequent change in the representation of the world is called *updating*.
- We have an incomplete, approximate or erroneous representation of a static world. The new information represent a new evidence regarding this world. The consequent change in the representation is properly called *revision*.

In our scenario, a loosely-coupled group of peers are incrementally building an ontology for a fixed domain. Thus, the world is not supposed to change, while the world's description is constantly evolving as the participants add, refine or retract classes and properties definitions. This scenario is that of a *revision* process and need to be handled within a framework possessing some specific requisites. The need for those requisites already appeared in a juridical scenario (incremental building of a proof in court[9]) and in distributed multi-agent belief revision[21].

Ability to reject incoming axioms. A belief revision system for a multi-source environment should drop the rationality principle of "priority to the incoming information", which is not acceptable since the sources are asynchronous and there is no strict correlation between the chronological sequence of information and their credibility or importance[11].

The ability to recover previously discarded axioms. Each domain expert should be able to recover previously discarded pieces of the the ontology if new axioms redeem them. This should be done not only when the new axioms directly support previously rejected axioms, but also when they indirectly support them by disclaiming the axioms that caused their ostracism.

For these reasons we adapt to ontology revision a belief revision framework that replace the priority to incoming information with the *principle of recoverability*[10]. The rationale for this principle is that, if an axiom was part of the ontology in the past, and it would be consistent with the current ontology, then it should be part of the ontology again.

4 Revision procedure

Belief revision has been approached both as a qualitative syntactic process and as a numerical mathematical issue. Our distributed ontology revision system links symbolic and numerical techniques. Computationally, the ontology revision consists of two steps acting on the axioms of the ontology, and three steps working with numerical weights.

Each peer stores his knowledge about the domain in at least two repositories[10]:

1. A *background repository* KB. This is the set of all axioms and facts available to reasoning; it contains both the axioms and facts written by the contributor and received from other contributors. It may be inconsistent.
2. A *working ontology* $B \subseteq KB$, which is the maximally consistent, currently preferred ontology that should be used for reasoning or further editing.

Given an incoming contribution p (an axiom or a fact) from a source, the evolution process consists of the following steps:

1. detection of minimally unsatisfiable subsets of $KB \cup \{p\}$, called *nogoods*;
2. generation of the maximally satisfiable subsets of $KB \cup \{p\}$, called *goods*;
3. revision of the credibility weights of axioms in $KB \cup \{p\}$;
4. choice of a preferred maximally consistent subset of $KB \cup \{p\}$ as the new working ontology B';
5. recalculation of "a posteriori" reliability of sources.

4.1 Symbolic steps

Step 1 and 2 are symbolical ATMS-style operations[22]. We define a *nogood* as a minimally inconsistent subset of KB. Dually, we define a *good* as a maximally consistent subset of KB.

Nogood detection can be demanded to a reasoner, such as Racer[23], FaCT[24], Pellet[25]. The set of goods and nogoods are dual: if we remove from KB exactly one element for each nogood, what remains is a good[26]. So, once an inference engine finds out some nogoods, it is possible to use a set-covering algorithm, such as the one introduced by Reiter for model-based diagnosis[27], to find out the goods. This algorithm has already been succesfully used for belief revision[21].

An interesting property that the inference engine does not need to calculate the collection of all nogoods (i.e. *minimally* inconsistent subsets of KB), but just a collection of inconsistent subsets of KB, which is much easier.

4.2 Numerical steps

The numerical approach to ontology revision deal with the ontology as a set of weighted axioms. Weights usually are reals between 0 and 1, representing explicitly the credibility of the axioms.

The numbers represent uncertainty caused by the not complete reliability of the team members [3]. As the reliability of the source is strongly related to the credibility of the information, it is necessary to deal with couples ⟨source, axiom⟩[28].

The numerical steps of the revision procedure are step 3–5.

Step 3 of the ontology revision process uses the belief function formalism, as the one used by Shafer and Srivastava for auditing[29]. From the reliability value of each source (a propability that the source gives correct information), the credibility of the goods is determined by the Dempster rule of combination. Thus, ontology revision consists in the reassignment of credibility to axioms in the light of the incoming axiom. The credibility ordering reflects the collaborative building of the ontology: the reliability and the number of different contributors affect the credibility of the axiom and the converse.

The recalculation of credibility values involves all the collected axioms in KB. The incoming axiom p is confronted not just with the current ontology B, but with all KB, so that the weight of axioms in $KB \cup \{p\}$ are reviewed in a broader and less prejudicial basis.

Step 4 is the selection of a new ontology B'. The new ontology is the maximally consistent subset of $KB \cup \{p\}$ with the greater credibility. Since the incoming information causes a recalculation of all the credibility values, and the selected ontology is maximal, it is possible to rescue axioms from KB.

Even when the new contribution is compatible with the working ontology (meaning that $B \cup \{p\}$ is satisfiable), not necessarily $B' = KB \cup \{p\}$, since the global revision of numerical weight in step 3 may yield a totally different choice of ontology in step 4. A previously rejected set of axioms r can be rescued if p support r against a previously accepted set q.

In general, even when the new ontology B' is syntactically equal to the previous B, meaning that p has been rejected, B' may have a different credibility distribution (assignment of weights) from B. The incoming contribution p might be rejected even when a new ontology B', different from B, is selected, but $B' \cup \{p\}$ is still unsatisfiable.

Step 5 uses Bayesian conditioning to determine the probability that a source give correct contributions, gives the new accepted ontology B'. The main point is that a reliable source can not give false informations, while an unreliable source may occasionally give correct contributions.

As an alteration of the credibility of an axiom might result in the perturbation of the credibility of all the axioms from the same source, thus causing a completely different ontology to be selected at the next step.

[3] Even the contribution from the agent self can be considered not completely reliable, as this depends of the relative trust a contributor has on his work compared to trust on other experts' works.

5 Examples

We will show two examples, showing the symbolical and numerical steps respectively. In both, we suppose that a group of domain experts are working on an ontology of birds.

5.1 Symbolic Example

The initial knowledge base KB of one of those experts is made of the axiom $Bird \sqsubseteq Fly$ and the fact $Bird(Tweety)$, where $Bird$ and Fly are classes (the class of individuals that are birds, and the class of individuals that can fly, respectively), while Tweety is an individual. The knowledge base is consistent, so the initial ontology is $B = KB$.

The expert receive from a colleague (let's call him source 1) the axiom $\neg Fly(Tweety)$. Now $KB = \{Bird \sqsubseteq Fly,\ Bird(Tweety),\ \neg Fly(Tweety)\}$ is unsatisfiable. If we would adopt the AGM principle of Priority to Incoming Information, the new working ontology would be chosen among

1. $B_1 = \{Bird \sqsubseteq Fly,\ \neg Fly(Tweety)\}$
2. $B_2 = \{Bird(Tweety),\ \neg Fly(Tweety)\}$

If we adopt the Principle of Recoverability instead, we have a third candidate working copy,

3. $B_3 = B = \{Bird \sqsubseteq Fly,\ Bird(Tweety)\}$

Next, another expert (let's call him source 2) affirms $Fly(Tweety)$.

If we use the AGM principles, the new working ontology would be, respectively,

1. $B_{1'} = \{Bird \sqsubseteq Fly,\ Fly(Tweety)\}$, if B_1 was chosen after the input from source 1,
2. $B_{2'} = \{Bird(Tweety),\ Fly(Tweety)\}$, if B_2 was chosen.

If we allow the rejection of the new contribution, after the arrival of the axiom from source 2, we can:

1. Reject the new axiom. Our working copy remain the same as after step 1.
2. Accept the new axiom.
   (a) $B_1 = \{Bird \sqsubseteq Fly,\ \neg Fly(Tweety)\}$. We recover $Bird(Tweety)$, so $B_{1''} = \{Bird \sqsubseteq Fly,\ Bird(Tweety),\ Fly(Tweety)\}$.
   (b) $B_2 = \{Bird(Tweety),\ \neg Fly(Tweety)\}$. We recover $Bird \sqsubseteq Fly$, so $B_{2''} = \{Bird \sqsubseteq Fly,\ Bird(Tweety),\ Fly(Tweety)\}$.
   (c) $B_3 = \{Bird \sqsubseteq Fly,\ Bird(Tweety)\}$. This is a simple expansion, so $B_{3''} = \{Bird \sqsubseteq Fly,\ Bird(Tweety),\ Fly(Tweety)\}$.

The example show that, if we consider the axiom $Fly(Tweety)$ more credible than $\neg Fly(Tweety)$, our final working ontology would be the same, independently from the choice made at the first step.

5.2 Numerical Example

The initial knowledge base KB of one expert is made of the axiom $Bird \sqsubseteq Fly$ and the fact $Bird(Tweety)$.

The expert receives from source 1 the axiom $\neg Fly(Tweety)$ and chooses as the new working ontology $B_2 = \{Bird(Tweety), \neg Fly(Tweety)\}$.

Now suppose source 1 sends us the axiom $\neg Bird(Tweety)$. If we reject this axiom, probably now our confidence in source 1 will be lower, as the credibility of the information affects the reliability of its source[30].

A change on the credibility of an axiom provided by a source yields corresponding changes in the credibility of the other axioms provided by the same source, even if they are not logically related with each other. As a consequence of this perturbation, a completely different working ontology might be chosen, in the previous example B_3 instead of B_2, thus rejecting the previously accepted axiom from source 1. Since all the collected axioms are retained and their weights can change, the new selection might reconsider some previously discarded axiom, whether the incoming contribution is accepted or not.

Probably, the last come contribution decreases the credibility of the axioms it would render unsatisfiable, even in the case it has been rejected. The same when we receive an axiom which already belongs to the working ontology: it is not the case that nothing happened, as AGM fourth postulate of expansion would suggest[6, p. 49], since we are now, in general, more sure about the correctness of the axiom.

6 Conclusions and Future works

When a group of peer tries to capture in an ontology a static domain, but their domain's knowledge is only partial and potentially unreliable, not all the contribution can be taken as unconditionally useful. It is necessary to use an ontology revision procedure that allows to discard the incoming information, if there is no reason to consider it more reliable than other conflicting contributions, and to rescue previously discarded axioms, if they are now compatible with the current selected ontology.

In general, at each step there will be more than a consistent subset of the ontology with maximal size (i.e., a good). There is the need of a rational criteria to choose a good as the new working ontology. If we keep track of axioms' sources and give to each peer a a-priori reliability value, we can use the belief-function formalism to estimate the reliability of each good and bayesian conditioning to evaluate a new a-posteriori reliability value for the sources.

This work is just the beginning of an analysis of ontology revision process for a distributed environment. Current research work involves the following subjects.

Collaborative Ontology Revision At the end of the work each expert has its own version of the domain ontology. To extract the final result of the collective work of the group of interacting experts, a voting mechanism is needed. The

integration of the different conceptualizations must not be performed by an external supervisor, but it can be done by the group itself.

Ontology distribution To allow the distribution of individual fragments of the ontology, it must be possible to partition it and then reconstruct it preserving meaning.

For RDF, this bring to the definition of the *minimal self-contained graph*[31] as the finer decomposition of a graph that would preserve meaning. This minimal set consists in a statement and, recursively, all statements involving a blank node already in the set.

Given the OWL RDF/XML syntax's use of blank nodes to build complex definitions, a similar concept can be applied to OWL. This decomposition allows the distribution of the ontology between peers, as in the scenario introduced in section 1.

User interaction A software supporting the collaborative building of an ontology must be able to use an ontology revision mechanism to maintain a consistent working ontology. Where inconsistencies arise and there is no other available ranking, the choice among different maximally consistent subsets can only be done by the user.

However, there are other times during the work when an user intervention would be useful. Why don't allow the user to explicitly mark a part of the ontology as unreliable, not necessarily causing its deletion from the current working set, but determining a change in the distribution of reliability among the sources?

Explicit reliability judgments by an human agent must be taken into account when the system build a credibility ranking among the available sources.

Strong time-Independence Even if the new information has no priority for his novelty, a complete independence of axiom's weights from contributions' arrival time is not guaranteed. This, given the asynchronous setting, would be a desirable feature of the system.

Ontology versioning Ontology versioning is defined as the ability to handle an evolving ontology by creating and managing different variants of it[2]. A common requirement between ontology versioning and the present ontology revision framework is the ability to work with different versions of the ontology and to recover previous parts of it. Thus the revision process for ontology revision can be at some extent applied to ontology versioning.

References

1. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition **5**(2) (1993) 199–220
2. Klein, M., D.Fensel: Ontology versioning on the semantic web. In: Proc. of the Int. Semantic Web Working Symposium (SWWS). (2001) 75–91

3. Flouris, G.: On Belief Change and Ontology Evolution. PhD thesis, Dept. of Computer Science, University of Crete (February 2006)
4. Dou, D., McDermott, D., Qi, P.: Ontology translation on the semantic web. In: International Conference on Ontologies, Databases and Applications of Semantics. (2003)
5. Alchourrón, C.E., Gärdenfors, P., Mankinson, D.: On the logic of theory change: partial meet contraction and revision functions. The Journal of Symbolic Logic **50** (1985) 510–530
6. Gärdenfors, P.: Knowledge in Flux: Modeling the Dynamics of Epistemic States. MIT Press, Cambridge, MA (1988)
7. Ribeiro, M.M., Wassermann, R.: First steps towards revising ontologies. In: Proc. of WONRO'2006. (2006)
8. Hansson, S.O.: A survey of non-prioritized belief revision. Erkenntnis **50** (1999) 413–427
9. Dragoni, A.F., Nissan, E.: Salvaging the spirit of the meter-models tradition: A model of belief revision by way of an abstact idealization of responce to incoming evidence delivery during the construction of proof in court. Applied Artificial Intelligence **18** (2004) 277–303
10. Dragoni, A.F.: Belief revision: From theory to practice. The Knowledge Engineering Review **12**(2) (1997)
11. Dragoni, A.F., Mascaretti, F., Puliti, P.: A generalized approach to consistency based belief revision. In Gori, M., Soda, G., eds.: Topics in Artificial Intelligence, Proc. of the 4th Conference of the Italian Association for Artificial Intelligence. Number 992 in LNAI, Florence, Italy, Springer-Verlag (October 11–13 1995) 231–236
12. Bechhofeer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., Stein, L.A.: OWL web ontology language reference. Recommendation, W3C (10 February 2004)
13. Hayes, P.: RDF Semantics. Recommendation, W3C (February 10, 2004)
14. Baader, F., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: Description Logic Handbook: Theory, implementation and applications. Cambridge University Press (2002)
15. Patel-Schneider, P.F., Hayes, P., Horrocks, I.: OWL web ontology language semantics and abstract syntax. Recommendation, W3C (February 2004)
16. Heflin, J., Hendler, J.: Dynamic ontologies on the web. In: Proc. of the 17th Nat. Conf. on Artificial Intelligence, Austin, Texas (30 Jul – 3 Aug. 2000) 443–449
17. Foo, N.: Ontology revision. Lecture Notes in Computer Science **954** (1995) 16–??
18. Halaschek-Wiener, C., Katz, Y.: Belief base revision for expressive description logics. In: Proc. of the OWLED'06. (2006)
19. Qi, G., Liu, W., Bell, D.: Knowledge base revision in description logics. In: Proc. of 10th European Conf. on Logics in Artificial Intelligence (JELIA'06). Number 386-398, Springer Verlag (SEP 2006)
20. Haase, P., Stojanovic, L.: Consistent evolution of OWL ontologies. In: Proc. of the 2nd European Semantic Web Conf. (ESWC-05). (2005)
21. Dragoni, A.F., Giorgini, P.: Distributed belief revision. Autonomous Agents and Multi-Agent Systems **6** (2003) 115–143
22. de Kleer, J.: An assumption-based truth maintenance system. Artificial Intelligence (28) (1986) 127–162
23. Haarslev, V., Möller, R.: Racer system description. In: Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2001). Volume 2083 of Lecture Notes in Artificial Intelligence., Springer (2001) 701–705

24. Horrocks, I.: The FaCT system. In de Swart, H., ed.: Proc. of TABLEAUX98. Number 1397 in LNAI, Springer (1998) 307–312
25. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Technical Report 2005-68, University of Maryland, Institute for Advanced Computes Studies (UMIACS) (2005)
26. Roos, N.: A logic for reasoning with inconsistent knowledge. Artificial Intelligence **57** (1992) 69–103
27. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence **32** (1987) 57–59
28. Dragoni, A.F.: A model for belief revision in a multi-agent environment. In Demazeau, Y., Werner, E., eds.: Decentralized A.I. Volume 3., North Holland Elsevier Science Publisher (1992) 103–112
29. Shafer, G., Srivastava, R.: The bayesian and belief-function formalisms a general perspective for auditing. In Shafer, G., Pearl, J., eds.: Reading in Uncertain Reasoning. Morgan Kaufmann Publishers (1990)
30. Mazzieri, M., Dragoni, A.F.: On the relation between trust on input and reliability of output. In: Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006). (2006)
31. Tummarello, G., Morbidoni, C., Puliti, P., Piazza, F.: Signing individual fragments of an RDF graph. In: Proc. of the World Wide Web Conference (WWW2005). (2005)

Dynamic Ontology Co-Evolution from Texts: Principles and Case Study

Kévin Ottens[1], Nathalie Aussenac-Gilles[1], Marie-Pierre Gleizes[1], Valérie Camps[1]

[1] Institut de Recherche en Informatique de Toulouse
Université Paul Sabatier
118, Route de Narbonne 31062 Toulouse cedex 9 – France
{ottens/aussenac/gleizes/camps}@irit.fr

Abstract. As claimed in the Semantic Web project, a huge amount of physically distributed interacting software agents could find the semantic of available resources and answer more relevantly to users' requests if the content of these resources would be represented with formal semantic concepts defined in ontologies. Because Web information sources are highly dynamic and conceptually heterogeneous, one of the most challenging problems in the Semantic Web research is the proper and frequent ontology updating in keeping with knowledge changes. To tackle this problem, we have developed a self-organizing multi-agent system -Dynamo- able to create an ontology draft from automatic text processing. Because it is well-known that only a part of a domain description is explicitly described in texts, Dynamo enables an ontology co-construction with a domain expert in a fully interactive way. In this paper, we present the principles of this approach and related experiments.

Keywords: Collaborative ontology construction from text, adaptive multi-agent system, ontology dynamics, ontology maintenance.

1 Introduction

The challenge of an efficient information retrieval on the Web requires to define relevant resources for document tagging and indexing. Two apparently competitive trends emerged: whereas the Semantic Web [1] suggests the use of normalized and formal concepts in ontologies defined by domain specialists, the Web 2.0 tools make it possible to collaboratively organize and share hierarchies of possible tags. These two trends offer complementary features. Their combination could benefit both of the precision and formalism of ontologies, and of the fast reactivity and the powerful collaborative effort that lead to build Web2.0 lists of tags. Recent investigations propose to rely on the strengths of these two trends, mainly to get updated resources that match the evolution of knowledge sources on the Web.

Indeed, ontologies are rigid structures that are difficult to update. When used in Semantic Web applications, they are immerged in a highly dynamic environment, where new and conceptually heterogeneous information sources appear every day. Domain specific and technical knowledge is more prone to change than expected. An

attempt to evaluate this dynamics [2] has shown that ontology maintenance is now one of the key issues for their use in Web applications: "*The only feasible approach for dealing with dynamic domains is speeding up ontology maintenance. It is obvious that monthly or weekly updates of the ontologies in our simulation experiments will drastically reduce the amount of missing elements*". So far, one of the major challenges for the Semantic Web research is the proper and frequent ontology updating in keeping with knowledge changes.

These changes could come from the integration of tags list built in Web 2.0 collaborative applications, from the integration of new Web sites and databases, or from manual modifications proposed by experts. To tackle this problem, we propose to combine the recent advances in ontology learning from texts with the help of Natural Language Processing tools and the flexibility of adaptive agent programming. We have developed a self-organizing multi-agent system - Dynamo[1] - able to create and maintain an ontology draft from automatic text processing. As long as only a part of domain knowledge is explicitly described in texts, Dynamo expects domain experts to add missing knowledge to this draft and to interact with the system until they get a satisfying ontology. This system assumes that ontology engineering is a continuous cycle where texts or humans may suggest some modifications. In this paper, we present the principles of ontology co-construction with Dynamo and some validation experiments of the approach.

First, we briefly describe works related to ontology construction and maintenance from texts. Section 2 expounds the basic principles of the distributed Dynamo algorithm that creates a draft ontology from text. This algorithm is implemented with a multi-agent system where the agents are the concepts of the ontology running to discover their right place inside the organization. Section 3 illustrates with an example the process of ontology creation from text. This is a co-construction process where the ontologist and Dynamo interact in real-time according to their respective knowledge. Properties of this software are analysed with regard to this experiment in section 4 before concluding in section 5.

1.1 Ontology Engineering from Texts: Short Overview

Ontology engineering from texts has reached enough maturity to be considered as an efficient way to build ontologies, with the extra advantage that various lexical forms can be obtained for each concept. Recent books like [3] and [4] provide a good overview of existing methods and tools. They illustrate the diversity of techniques that can be applied to get various kinds of specific linguistic evidences of domain knowledge. These syntheses confirm the necessity to combine linguistic and statistical approaches to text mining with different perspectives, like term extraction, semantic class identification, relation extraction, … Whatever the quality of the tools and the relevance of their combination may be, only a part of an ontology can be learned from text: results of the learning process generally are called *draft* or *kick-off ontologies* [5]. They need to be formalized and their ontological properties have to be checked.

[1] DYNAMO is an acronym for « DYNAMic Ontologies »

Nevertheless, only a few methods, like Text2Onto [6], have paid attention to ontology maintenance by using Natural Language Processing. The Text2Onto framework helps to semi-automatically learn and update ontologies from domain specific texts by applying machine learning techniques [6]. [7] uses a neural network system for term extraction and latent segment analysis for term clustering and incremental concept identification. In both cases, the authors underline the need for these tools to provide facilities for a manual engineering of the learned network. Only human intervention can guarantee that the ontology fulfils the application requirements. Maedche and Staab call it *balanced cooperative modelling* [8].

1.2 Statements underlying Dynamo

Our contribution follows this paradigm. Our system, Dynamo, can be used to build ontologies or to maintain them. The current system is able to maintain only Dynamo designed ontologies. But the target is to be able to dynamically update existing models with the knowledge learned from texts. We focus mainly on term extraction as a means to identify domain concepts, and on term clustering based on their syntactical structure to learn hierarchical relations. In our approach, term extraction is carried out by an independent tool, the Syntex system [9], that runs syntactical and distributional analyses. Dynamo defines an adaptive multi-agent system (MAS) from each terminological network provided by Syntex and the available agents that form the ontology to be maintained. These agents organise themselves so that they form a hierarchy of concepts. We consider this hierarchy as the resulting draft ontology. Because it combines a conceptual network and related terms, we call it a *termino-ontological resource*.

The organization process relies on a clustering algorithm, detailed in [10], the originality of which is to be distributed over all the agents. Although its design is inspired by classical agglomerative hierarchical clustering [11], this algorithm tends to break up clusters locally identified by each agent. Inputs are the candidate terms provided by Syntex, and it exploits syntactical relations between terms to define clusters. The major gain brought by this new implementation is that feed-back can be manually provided before the clustering is completed, which makes it possible to understand and modify the obtained clusters. This MAS enables the dynamic construction of a class hierarchy from an entry data flow. Each node of the hierarchy is a concept-agent created when a new term is taken into account. An agent's behaviour enables to merge it with a sibling agent or to raise one of its child agents, according to a similarity measure locally computed. The resulting classification is the hierarchy of the multi-agent system itself. As we will see in section 2, this agent's behaviour is not sufficient to create an ontology or even a taxonomy for two reasons: there is no rule to simplify the hierarchy and this is no multi-criteria algorithm.

2 Ontology as a Self-Organizing Multi-Agent System

Dynamo is a tool, based on an Adaptive Multi-Agent System (Amas), enabling the construction and the maintenance of an ontology starting from a textual corpus. Multi-

Agent Systems provide solutions to problems involving several autonomous entities (called "agents") which can be geographically and logically distributed, which are plunged into a dynamic environment, which have a partial perception of this environment, and which have limited cognitive capacities. More precisely, the aim of Dynamo is to build a draft domain specific termino-ontological resource (the multi-agent system or MAS). This draft is a hierarchy of concepts which results from the MAS organisation where each concept is represented by an agent. Dynamo is a semi-automatic tool because the ontologist[2] has to validate, refine or modify the hierarchical organization of concepts until it reaches a satisfying state. The Dynamo system consists of three parts:

- a network of terms, obtained with the Syntex term extractor [9] from a textual corpus. Syntex runs a dependency structure analysis to extract all possible candidate-terms from a corpus (in French or in English); it relies on head-expansion relations between compound terms to organize them into a network[3]; and it runs a distributional analysis in order to suggest classes of terms that share similar syntactic contexts. Each term is given in its lemmatized form, with a list of all its occurring sentences, its head term and expansion terms, related terms that share similar use contexts and statistics (frequency, productiveness …). Syntex has been used many times for ontology building;
- a multi-agent system which carries out a hierarchical clustering over the term network and produces a taxonomy of concepts. Agents composing the system cooperate to position themselves in a hierarchy and the multi-agent system constitutes the resulting taxonomy. When building a net ontology, this network is empty at first, but when maintaining an ontology built with Dynamo, it contains the current network of concept-agents.
- an interface enabling the ontologist to visualize and control the clustering process, and to modify the resulting hierarchy.

Our approach to create an ontology as the result of self-organising process in a MAS is, to our knowledge, completely original. "*Self-organisation is the mechanism or the process enabling a system to change its organisation without explicit external command during its execution time*" [12]. This choice comes from the qualities offered by this kind of multi-agent systems: they make easier the interactive design of a system (in our case, a conceptual network), they enable its incremental building by progressively taking into account new data (coming from text analysis and user interaction), and, last but not least, they can be easily distributed across a computer network. With this approach, ontology is seen as a stable network composed of conceptual entities, here represented by "concept agents", linked with labelled relations. Another advantage over a centralized clustering algorithm is that results of intermediate steps can be checked and corrected.

The distributed clustering algorithm implemented with an Amas (whose principle and evaluation are explained in [13]) tends to introduce new layers in the taxonomy. It

[2] We call an *ontologist* a knowledge engineer or an analyst, in charge of building an ontology from knowledge sources.

[3] For instance, the term *knowledge acquisition* has the term *acquisition* as head and *knowledge* as expansion, and it is the expansion of the term *knowledge acquisition system*.

is designed to be both the system producing the resulting structure and the structure itself. It means that each agent represents a concept and its autonomous and cooperative behaviour is to find its right place in the organization, namely in the taxonomy. Each agent possesses communication capacities and behaviours to structure and modify the taxonomy according to different rules. The system output is the organization obtained from the interaction between agents, while taking into account feedback coming from the ontologist when he modifies the taxonomy according to the application requirements or his expertise.

Furthermore, the agents' behaviour rules enable several organisational modifications locally by taking into account their parents/child relations. These local modifications are listed in the three following points:

1. The "head coverage" rule tends to push involved agents toward the leaves of the taxonomy. To do that, each agent determines if its parent is adequate. This is possible because each concept agent is described by a set of terms that belong to the head-expansion term network. If T_X is the set of terms describing a concept agent X and head(T_X) the set of all the terms that are head of at least one item of T_X, the parent adequacy function a(P,C) between a parent P and a child C can be defined by the following formula : $a(P,C) = |T_P \cap head(T_C)| / |T_P \cup head(T_C)|$. Then, the best parent for C is the agent P that maximizes a(P, C).
   Rule1: when agent C is unsatisfied with its parent P, it evaluates a(B_i, C) with all its siblings (noted B_i); the one maximizing a(B_i, C) is chosen as the new parent.

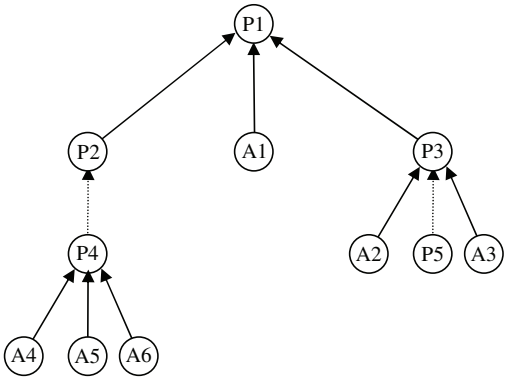


Figure 1. Simplification branch and uselessness rules

2. The "simplification branch and uselessness" rules force the agent to go up the hierarchy, as shown in the figure 1.
   Rule2: When an agent has several children but no sibling (like P_4), then it proposes to its children (A_4, A_5 and A_6) to have its own parent (P_2) as new parent.
   Rule3: When an agent has no children and is represented by no term (like P_5), it has to leave the system.
3. The "similitude tolerance" rules enable to obtain n-ary trees forcing the agent to go up the hierarchy and to simplify the structure by aggregation. More precisely, with

the distributed clustering algorithm and the previously presented rules, the result of the Amas is necessarily a binary tree (unless for the last level of the hierarchy if the rule 1 has been applied). The hierarchy resulting from this basic algorithm is always a binary tree because this algorithm separates items when similarity is different from 1.0 in order to form clusters. But our objective is to obtain a dynamic taxonomy, which is rarely a binary tree. To obtain n-ary nodes rather than binary nodes, each concept agent A introduces a tolerance ε in its vote (which follows the Condorcet vote strategy), and only keeps its vote for its siblings F_k such as $1 - sim(A, F_k) > \varepsilon$. This tolerance is locally managed by each concept agent; it takes into account the tolerance value of its parent and its own tolerance in order to influence the connection factor. The ontologist can give to the system an interval for the global connection factor and each concept agent has then to adjust its local tolerances to try to conform to this interval while taking into account dissimilarities with its neighbourhood. Two rules have been defined to take into account these tolerance variations.

Rule4: When an agent P_0 has its children which do not enforce any more the property about tolerance ε_{P0}, then P_0 proposes to its children to have its parent P as new parent.

Rule5: When an agent P_0 has a number of children too high (resp. too low), it decreases (resp. increases) its tolerance ε_{P0}.

Each concept agent has to deal with multiple criteria during the taxonomy building and has to determine its priorities at a given time. More precisely, each concept agent computes three non cooperation degrees and chooses its current priority according to the highest one. This priority is used during message passing and each message possesses a priority p_k corresponding to the non cooperation degree of the agent when it sends it. For an agent A having a parent P, a set of siblings B_i and which received a set of messages M_k having the priority p_k, the three computed non cooperation degrees are:

- $\mu_H(A) = 1 - a(P, A)$, is the "head coverage" non cooperation degree, determined by the head coverage of the parent,
- $\mu_B(A) = \max(1 - \text{similarity}(A, B_i))$, is the "siblings" non cooperation degree, determined by the worst sibling of A regarding similarities,
- $\mu_M(A) = \max(p_k)$, is the "message" non cooperation degree, determined by the most urgent message received.

The non cooperation degree of agent A is $\mu(A) = \max(\mu_H(A), \mu_B(A), \mu_M(A))$. Then, we have three cases determining which kind of action A will choose:

- if $\mu(A) = \mu_H(A)$ then A will use the head coverage rule (rule1) previously detailed;
- if $\mu(A) = \mu_B(A)$ then A will run the distributed clustering algorithm [13];
- if $\mu(A) = \mu_M(A)$ then A will process M_k immediately in order to help its sender.

Those three cases summarize the current activities of the agents: they have to find the best parent for them when $\mu(A) = \mu_H(A)$; they have to improve the agent network structure through clustering when $\mu(A) = \mu_B(A)$; or they have to process other agent messages when $\mu(A) = \mu_M(A)$ in order to help them to fulfil their own goals.

In this approach, we consider an ontology as a dynamic equilibrium between its concept agents. The ontology modification is a perturbation of the previous

equilibrium caused by the appearance or disappearance of concept agents or relationships. In this way, a dynamic ontology is a self-organizing process occurring when new texts are included into the corpus, or when the ontologist interacts with it.

3 Co-Evolution Experiment

Dynamo has been experimented to create an ontology draft from a corpus of abstract scientific English papers published in the journal «Astronomy and Astrophysics» edited by Springer (http://www.springerlink.com/content/300419/). An ontology had been created from this corpus by experts and a sub-part of it (approximately one hundred nodes) will be the reference for evaluating our work. Because both the reference ontology and the learned one are designed from the same documents, we assume that their terminologies are overlapping. The experiments aim at showing the dynamic evolution relevance of the MAS. The set of parameters used during this experiment is composed of three elements:

1. The formula given in [14] to compare the similarity between two terms t1 and t2. This formula uses a, b, c and d which are respectively the number of contexts in which t1 and t2 are both present, only t1 (respectively only t2) is present and contexts where neither t1 nor t2 is present. The parameter α giving the balance between these contexts is fixed to 0,75.

$$sim(t1,t2) = \alpha\ /2\ *\ (\ a/(a+b)\ +\ a/(a+c)\ )\ +\ (1-\alpha)/2\ *\ (\ d/(d+c)\ +\ d/(d+b)\ )$$

2. The branching factor given by an interval [minValue, maxValue] which defines the number of children that a given concept could have in the graph. For example, if the maxValue is 2, we obtain a binary tree. According to his knowledge about astronomy, the ontologist has defined the branching factor interval as [2, 7].
3. We must also compare the obtained ontology from the experiment with the reference ontology. We used the measure of taxonomy overlapping given in [8] which defines a value between 0 and 1. When two taxonomies are identical, the corresponding measure is 1. This measure takes into account hierarchical relations and assumes that concepts with the same label are identical. So a low score of this measure means that the structure of both ontologies is very different.

3.1 Automatic Draft Ontology Creation from the Corpus

The system initializes the graph ontology by creating its root with the agent called TOP. Each term extracted from the corpus is then embedded into a corresponding agent concept linked with TOP. From this initial network of agent-concepts, the behaviour of each agent is launched according to the rules defined in section two. Each agent behaves in parallel by processing the local information coming from its neighbours. The self-organizing process of agents leads to a global equilibrium which corresponds to the initial draft ontology.

The result shown in figure 1 is then presented to the ontologist (for visibility reasons, we have suppressed some leaf concepts). Five main subsets found by

Dynamo have been highlighted with grey-blue dotted lines. They are identified as a root group, a main branch and three sub-branches.

The modifications carried out later on by the ontologist (see sections 3.2 to 3.5) will be considered as local perturbations by the concerned concept-agents. These agents will use again their behaviour in order to find a more cooperative location inside the organization (the ontology).

Figure 1. Resulting taxonomy from an autonomous resolution on the astronomy corpus

3.2 First Ontologist's Intervention

The similarity measure between the resulting taxonomy and the reference one gives a value of 0.78. This quite high value is mainly due to the good location of leaf concepts. Nevertheless, the global structure is unbalanced for the ontologist because the root graph contains three groups (sub-branch 1, 2 and 3) without clear semantics. Consequently the ontologist modifies the organization in bringing back these three groups (corresponding to ConceptAgent93, ConceptAgent94 and ConceptAgent97) to the root (he links them to the TOP concept hidden inside the "groupe racine").

These perturbations lead to a reaction of the considered concepts that re-evaluate their cooperation degree with their neighbours. Dynamo does about thirty link modifications, which leads to the new ontology draft shown in figure 2.

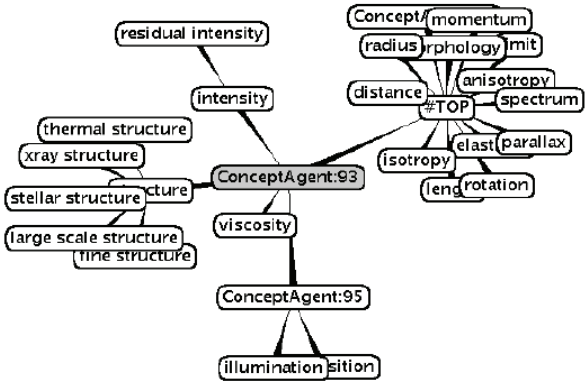


Figure 2. Resulting taxonomy after the first reorganization

In reaction to the changes made by the ontologist, Dynamo carries out the more important changes around ConceptAgent:93. These relevant changes enable to identify thematically ConceptAgent:93 as representing mechanical and thermo-dynamical properties. Moreover, the geometrical properties are directly linked to the TOP concept. This new structure obtains a value of 0.80 when compared to the reference ontology.

3.3 Second Ontologist's Intervention

Now the ontologist wants to improve the separation of the different emergent properties during his second intervention. His work consists in connecting thirteen concepts related with mechanical and thermo-dynamical properties to ConceptAgent:93. The concerned concepts are isotropy, morphology, momentum, anisotropy, spectrum, elasticity, mass, sensitivity, emission, density, entropy, diagram and temperature.

According to the behaviour rules, Dynamo moves then thirty concepts in the organization. The resulting structure contains now a complex branch describing the mechanical and thermo-dynamical properties. These actions by the ontologist moved the ontology away from the reference one and the similarity measure falls down to 0.76; this value will remain constant until the end.

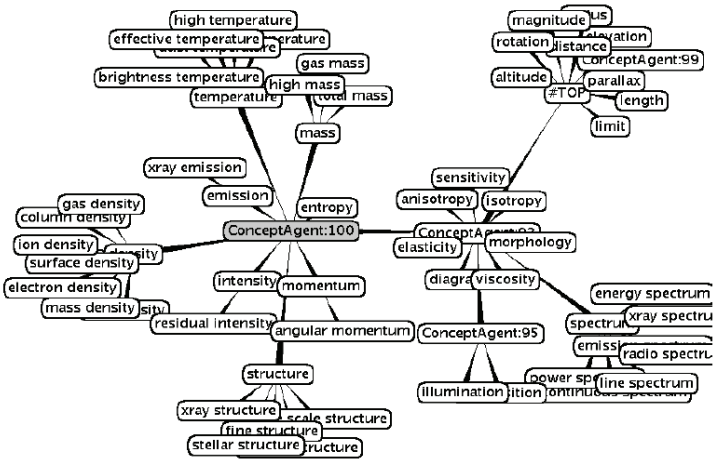


Figure 3. Organization after the second interaction between the ontologist and Dynamo

3.4 Third Ontologist's Intervention

The ontologist focuses now his work on the geometrical and optical properties found under ConceptAgent:99. He moves all the optical properties under the TOP concept, whereas some geometrical properties are linked to ConceptAgent:99.

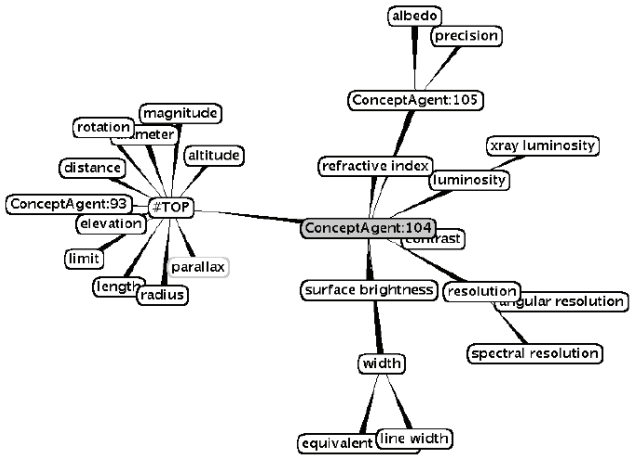


Figure 3. Third adjustment of the ontology

Dynamo reacts by moving all the geometrical properties under TOP and creates a new branch under TOP containing the optical properties and the "width" concept. This corresponds to a dozen of reorganizations processed by Dynamo.

The ontologist agrees with the destruction of the geometrical branch because he considers that the concepts previously aggregated had a high disparity. Thus, he keeps this change where geometrical properties were brought closer to thermo-dynamical and optical properties. Now, three coherent sets of concepts are linked to TOP: (*i*) the one directly connected under TOP; (*ii*) a group under ConceptAgent:93; (*iii*) a group under ConceptAgent:104.

3.5 Fourth Ontologist's Intervention

The remaining problem is the presence of the "width" concept under the optical branch (ConceptAgent:104). Thus, the ontologist moves it directly under TOP.

After the last self-organizing process of Dynamo (around ten link changes), we observe on the left a group about mechanical properties, in the middle a group about geometrical properties, and the optical properties on the right.

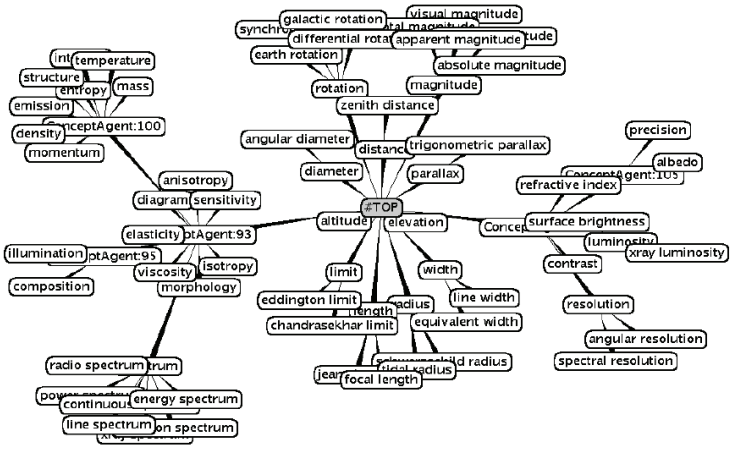


Figure 4. Final co-evolving draft ontology

4 Evaluation Analysis

In this section, the evaluation is twofold: a quantitative evaluation which relies on performance results and a qualitative one which is made by the ontologist. We then give the main perspectives of this work.

4.1 Quantitative Evaluation

As explained with details in [13], the complexity measures of the Dynamo system give interesting results. We carried out this work roughly in two phases. Firstly, we determined that the theoretical complexity of the core distributed clustering algorithm is $\mathcal{O}(n^3)$ like the simplest centralized clustering algorithm. But, in practice our experiments showed a $\mathcal{O}(n^2 \text{Log } n)$ complexity on average with a very good stability of the system due to its low behaviour variation across our data sets.

Secondly, we computed the average complexity of the whole system (that is the core clustering algorithm and the extra rules presented in this paper). Then the experimental complexity raised to $\mathcal{O}(n^3)$, once again with a very low variation and then a good stability of the system. Of course, this raise in complexity is explained by the more refined result obtained as output of the system. The system does more computations in this case, but the complexity of the whole still stays acceptable.

4.2 Qualitative Evaluation

The time spent by the ontologist to co-construct the final draft ontology is around three hours, including the great part needed for the difficult handling of the visualization interface. He estimates that using a traditional tool would have required up to four times the time he spent with Dynamo. The number of modifications that he brought to the taxonomy is quite manageable: 3 during the first step, 13 during the second step, about 15 during the third one and only one at the last step.

The resulting taxonomy is even more refined than the reference one, which explains the sub-optimal similarity measure. Moreover, Dynamo reduces the cognitive load of the ontologist because he may focus on the hardest difficulties. Once he has modified these concept-agents, the system propagates the consequences of these changes on related agents. For example, in the experiment, Dynamo has relevantly modified the edges in the graph five times more than the ontologist did.

A possible new experiment could be to follow the method proposed in [15] to improve the evaluation of the learned ontology with regard to the reference ontology.

4.3 Improvement and Future Work

As in any software prototype, several features of Dynamo could be improved, mainly if we want to update existing ontologies in Dynamo. We will focus here only on the two most important ones: user-friendliness and link labelling.

The first limitation of Dynamo comes from the lack of user-friendliness of the end-user interaction. Even with a restricted ontology size, the ontologist has great difficulties in following the dozens of link modifications done autonomously in only few seconds at each step. He must spend a lot of time localizing in the graph display the concepts that he has previously worked on. Indeed, only a small perturbation made by the end-user can potentially have important repercussions on the structure. An efficient ontology maintenance system would require a deep cooperation with ergonomists to define an easy-to-use graphical interface.

The second limitation comes from the created links which are labelled only with "is-a". Consequently, the current Dynamo prototype produces taxonomies and not full ontologies. The main reason for this is that we focussed in priority on head-expansion relations between terms, and their most frequent meaning is a hyperonymy relation between a term and its compound terms. But the pre-processor Syntex is able to provide some linguistic clues to define other semantic relations and their labels. Moreover, results from other natural language processing tools could be given as input to the agents. In the near future, we envisage two complementary techniques to select proper labels for relations between concepts (or links among agents):

– The instantiation of a predefined set of patterns (for example [X 'take' Determiner Adjective Y]) defined by the ontologist in a given domain. These patterns could be used by pattern-agents inside Dynamo and their work would be to scan the taxonomy in order to fill in these empty generic patterns with relevant candidates.
– The automatic creation of these patterns based on the correspondence between the relationships between terms (given by Syntex) and links in the ontology which are not labelled yet. This allocation problem will be solved by using the Amas technology.

5 Conclusion and Perspectives

Ontology maintenance is a challenge that we propose to consider in continuity with ontology construction. In this paper, we presented a new approach based on multi-agent technology in order to reduce the ontologist's amount of work, by creating autonomously concepts and their relationships from text extracted candidate terms. Dynamo is an innovative tool for dynamic ontologies from two points of view:

- First, at any time, new sets of documents can be added to the input corpus, new knowledge can be manually provided by the ontologist, leading to concept and relation additions or deletions. The system adapts the previous network according to this new information; thus, the ontology can be effectively dynamically updated.
- Second, the system and the ontologist modify the same network in a cooperative way: this process relies heavily on the strong coupling between the action of one of them and the reaction of the other.

The semi-automatic ontology construction from texts eases greatly the ontologist's work. Nevertheless, based on our experience, there are a lot of implicit relationships which cannot be discovered in analyzing a corpus of texts. Thus, even in increasing greatly the computer work, the final decision remains to the human [16]. For this reason, we agree with the design requirements for ontology evolution defined by [17]:

1. *It has to (i) enable resolving the given ontology changes and (ii) ensure the consistency of the underlying ontology and all dependent artifacts;*
2. *It should be supervised allowing the user to manage changes more easily;*
3. *It should offer advice to user for continual ontology refinement.*

We think that a collective agent process -like Dynamo- is a good way to be consistent with these requirements.

References

1. Benjamins, V.R., Contreras, J., Corcho, O., Gómez-Pérez A.: Six Challenges for the Semantic Web. *Proceedings of the Semantic Web workshop held at KR-2002*, Toulouse, France (2002)
2. Hepp, M.: Ontology Maintenance: Quantifying the Conceptual Dynamics in Domain Ontologies. *DERI Technical Report SEBIS-TR2007-01* (2007)
3. Buitelaar, P., Cimiano, P., Magnini, B., editor(s), *Ontology Learning from Text: Methods, Evaluation and Applications*, Frontiers in Artificial Intelligence, (123) IOS Press,2005
4. Cimiano, P., *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer-Verlag New York, Inc.,Secaucus, NJ, USA,2006
5. Sure, Y., Staab, S., et Studer R.: Handbook on Ontologies, chapter 6. On-To-Knowledge Methodology (OTKM). Springer (2003).
6. Maedche, A., Volz. R., The Ontology Extraction and Maintenance Framework Text-To-Onto. *Proceedings of the conference on Data management and Knowledge Management*, DM_KM (2001)
7. Gargouri, Y., Lefebvre, B., Meunier, J.-G.: Ontology Maintenance using Textual Analysis. Proceedings of the Seventh World Multi-Conference on Systemics, Cybernetics and Informatics (SCI). Orlando, USA (2003)
8. Maedche, A., Staab, S.: Measuring Similarity between Ontologies. *Conference on Knowledge Engineering and Management (EKAW 2002),* pp 251–263. Springer-Verlag (2002).
9. Bourigault, D., Fabre, C., Frérot, C., Jacques, M.-P., Ozdowska S. : Syntex, analyseur syntaxique de corpus, in *Actes des 12èmes journées sur le Traitement Automatique des Langues Naturelles*, Dourdan, France. (2005)
10. Van Dyke Parunak, H., Rohwer, R., Belding, T. C., Brueckner, S.: Dynamic decentralized any-time hierarchical clustering. *29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval*. (2006)
11. Manning C. D. and H. Sch¨utze. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, Massachusetts, (1999).
12. Di Marzo Serugendo, G., Gleizes, M.-P., and Karageorgos, A. : Self-Organization and Emergence in Multi-Agent Systems. *The Knowledge Engineering Review* Vol.20, N°2, pp165-189 (2005).
13. Ottens, K., Gleizes, M.-P., Glize, P.: A Multi-Agent System for Building Dynamic Ontologies. *International Joint Conference on Autonomous Agents and Multi-agent Systems* (AAMAS 2007). ACM Press, p. 1278-1284 (2007)
14. Assadi, H.,: Construction of a regional ontology from text and its use within a documentary system. *Proceedings of the International Conference on Formal Ontology and Information Systems - FOIS'98* pp236-249. Trento, Italy (1998)
15. Dellschaft, K., Staab, S., On How to Perform a Gold Standard Based Evaluation of Ontology Learning. In *Proc. Of the International Semantic Web Conference 2006*: 228-241 (2006).
16. Brewster, C., Ciravegna., F., Wilks, Y.: Background and Foreground Knowledge in Dynamic Ontology Construction: Viewing Text as Knowledge Maintenance. *Proceedings of the Semantic Web Workshop,* SIGIR (2003).
17. Stojanovic, L., Maedche, A., Motik, B., Stojanovic, N.: User-Driven Ontology Evolution Management. *European Conference on Knowledge Engineering and Management (EKAW 2002*), pp. 285-300. Springer-Verlag (2002).

Extreme Tagging: Emergent Semantics through the Tagging of Tags

Vlad Tanasescu[1], Olga Streibel[2]

[1] Knowledge Media Institute, The Open University,
Walton Hall, Milton Keynes, MK7 6AA, UK
v.tanasescu@open.ac.uk, vladtn@gmail.com

[2] Netzbasierte Informationssysteme, Freie Universität Berlin,
14195 Berlin, Germany
streibel@inf.fu-berlin.de, ostreibel@gmail.com

Abstract. While the Semantic Web requires a large amount of structured knowledge (triples) to allow machine reasoning, the acquisition of this knowledge still represents an open issue. Indeed, expressing expert knowledge in a given formalism is a tedious process. Less structured annotations such as tagging have, however, proved immensely popular, whilst existing unstructured or semi-structured collaborative knowledge bases such as Wikipedia have proven to be useful and scalable. Both processes are often regulated through social mechanisms such as wiki-like operations, recommendations, ratings, and collaborative games. To promote collaborative tagging as a means to acquire unstructured as well as structured knowledge we introduce the notion of *Extreme Tagging*, which describes systems which allow the tagging of resources, as well as of tags themselves and their relations. We provide a formal description of extreme tagging followed by examples and highlight the necessity of regulatory processes which can be applied to it. We also present a prototype implementation.

Keywords: semantic web, web2.0, tagging, emergent semantics, meaning, semantic associations, knowledge paths.

1 Introduction

The process of building "a new brain for humankind" [1] as foreseen by semantic web research appears to be a slow one. Indeed, the semantic web contributed to the success of the notion of *ontology*, "a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualisation of the world" [2], but, possibly due to lack of software support [3], ontologies are difficult to build, even at the community level. Moreover, the final aim of the semantic web – data integration through ontology matching – is still a research question as it can be automated only in simple cases. Indeed, although there are already a large number of RDF files on the web, whether manually or automatically generated, only about 25 000 documents [4] representing semantic models, i.e. ontologies, are avail-

able online. This should not be the case, as ontologies should be easy to produce by each community, then shared in order to be aligned with others using the stack of specifications and languages – the semantic web "layer cake" [5] – designed to support this task [6].

In practice however, building and matching ontologies, appears to be an expert task, and difficulties related to knowledge acquisition, experienced decades ago in the artificial intelligence community, resurface. Moreover, while ontologies seem well suited to the description of scientific domains such as medicine and biology which are already semi-formal and organized by categories and part-of relationships, some communities such as geospatial scientists only accept with scepticism the exclusive usage of ontologies to describe their domains [7]. Arguments in favour of using alternative knowledge representation models include, amongst other, the inadequacy of category based reasoning to represent reality [8], the absence of grounding of symbolic systems [9], the need of different representations of the same entity according to the context [10], as well as the difficulty to represent psychological concepts such as *affordances* in a hierarchical way [11]. Indeed, we are still waiting for ontologies to be flexible enough to match the representational complexity of the human mind.

In the meantime, so called Web2.0 applications, by motivating users to contribute information, introducing fine tuned social regulation mechanism, as well as providing friendly user interfaces, have been experiencing both phenomenal growth and success. With the advent of Web2.0 the usage of unstructured annotations such as *tagging*, spread widely. Although the relation of tagging and social interaction has not, to our knowledge, been investigated in the literature, it seems to be the only way to allow users to describe their own content, since the system cannot determine in advance what this content will be. Collaborative tagging systems, by renouncing the use of predefined vocabularies, provide a simple way for users to give their own meaning to their own content [12].

Therefore, while current research is still trying to alleviate problems related to the practical use of ontologies, the semantic web may benefit from techniques used by Web2.0 applications. We believe that for the semantic web to expand faster, new semantic acquisition approaches, distinct from the centralized ontology development by experts, need to be explored. We also believe that any successful solution will use the social lever which raised the Web and Web2.0 to that level of popularity and usage.

Therefore, we introduce the notion of *Extreme Tagging Systems* (ETS), as an extension of collaborative tagging systems allowing the collaborative construction of knowledge bases. An ETS offers a superset of the possibilities of collaborative tagging systems in that they allow to collaboratively tag the tags themselves, as well as relations between tags. Unlike previous research on emergent semantics of collaborative tagging systems, ETS are not destined to exclusively produce hierarchical ontologies but strive to allow the expression and retrieval of multiple nuances of meaning, or semantic associations. The production of relevant semantic associations can then be automatically controlled through social network regulation mechanisms.

We first describe collaborative tagging systems. Then show the modifications introduced by extreme tagging systems, providing a formal definition. Accordingly, we explain our prototype implementation, and, before concluding, give some examples of regulation mechanisms that should be applied to the system.

2 The Semantics of Collaborative Tagging Systems

Collaborative tagging systems (CTS) support multiple users in the activity of tagging, which is marking content for future navigation, filtering or search [13]. As there is no prior agreed structure or shared vocabulary CTS users need neither prior knowledge nor specific skills to use the system [14]. We prefer to avoid the term folksonomy [15], not only because it is ambiguous (as stated in [13]), but also because of the relation to taxonomy, which seems to us unjustified in that context.

Tagging systems can be represented as hypergraphs [16] where the set of vertices is partitioned into sets:

$$U = \{u_1, \ldots, u_k\}, R = \{r_1, \ldots, r_m\}, \text{ and } T = \{t_1, \ldots, t_l\} \ . \quad (1)$$

U, *R*, and *T* correspond to *users*, *resources*, and *tags*. An annotation, i.e. a resource tagged with a tag by a user, is an element of set A, where:

$$A \subseteq U \times R \times T \ . \quad (2)$$

The final hypergraph formed by a collaborative tagging system is defined as *G* with:

$$G = \langle V, E \rangle \text{ with vertices } V = U \cup R \cup T, \text{ and edges } E = \{\{u, r, t\} \mid (u, r, t) \in A\} \ . \quad (3)$$

Collaborative tagging systems have proved extremely popular. Their strengths consist in generating *serendipity* while browsing – the fact of being able to retrieve what others have tagged in a similar way, e.g. one can retrieve everything that has been annotated using the tag "ant" –, as well as the elaboration of *desire lines* – a non constrained reflection of the user's vocabulary – through a dataset [12] (e.g. I can use the English tag "ant" or the French "fourmi" indifferently, without being constrained by the system). However, when compared to more formal descriptions of domains, CTS are criticized for their *ambiguity* (an "ant" tag may be found for a resource related to "Actor Network Theory", the "Apache Ant project", or a representation of the insect), the dealing with *multiple words* constituting a single tag ("semantic web", "semanticweb" or "semantic-web" for example) or *synonymy* ("mac" "macintosh", and "apple"). These issues have leaded some to colloquially describe tagging systems as "a mess".

To go toward "less mess", approaches have been proposed to find groups of related tags by using tag co-occurrence for given resources [17][18][19]. Moreover, most websites using collaborative tagging systems already present tag clouds – a representation of a resource's annotations where each tag is visually weighted by his number of occurrences –, or allow presentation by tag clusters – several tags are grouped under an appellation – and often offer tag recommendations – tags are suggested according to previous annotations.

Furthermore, some semantic web oriented approaches attempt to extract ontologies from collaborative tagging systems. In [16] the author maps tags onto concepts and resources to instances and applies network analysis techniques to cluster them. [20] presents an ontology for tags which would allow them to be shared and exchanged be-

tween systems, while in [21] the authors mine association rules between tagged resources to recommend tags, users, or resources, discovering supertag relations as well as resource communities. In [22] the authors deduce clusters and relations between tags by relating them to background knowledge obtained through ontology searches while [14] presents an experiment to automate the previous method.

Ultimately, ontologies and tagging systems are both symbolic frameworks, and as such they are subject to the criticism of the lack of a retrievable grounding. Indeed, in both cases by using symbols (in a given language), the expressed concept, or *signified*, remains in people's minds, and the resulting symbol networks may appear – especially to a machine – as "free floating island[s] of reeds [with] no anchor in reality" [9]. However, CTS usually tag existing resources, i.e. specifying the referent, or ground, that symbols denote, without indicating the details of this denotation, as opposed to ontologies which first have to describe a domain before adding instances, and limit the grounding to a few pre-existing relations, i.e. the ones defined in the ontology (e.g. "part-of") plus the one assumed by the model (e.g. "is-a", "subclass-of", etc). Extreme tagging, by allowing the tagging of tags as resources as well as the specification of the relations between tags, is an attempt to push symbolic annotation frameworks to an extreme in order to see what the grounding problem becomes when any relation can be symbolically described at an arbitrary level of granularity.

3 Extreme Tagging Systems

An Extreme Tagging System (or ETS) offers a superset of the possibilities of collaborative tagging systems in that it allows to collaboratively tag the tags themselves, as well as relations between them. For example, a media resource representing the close-up of a car may be tagged with "car", "wheels" and "travel". The tag "wheels" itself may then be tagged (possibly by a different user) with "car" and "wheel", and the tag "car" itself could further be tagged with "vehicle" (cf. Figure 1)[1].

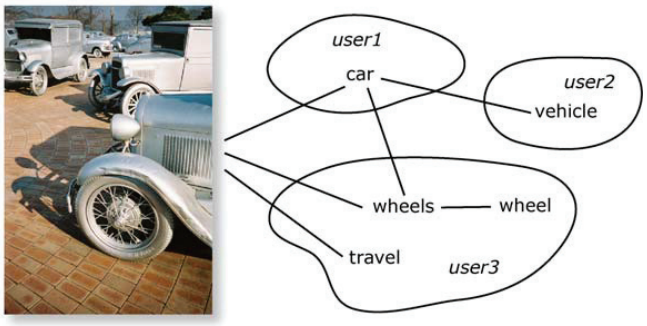


Figure 1. Tagging the tags.

[1] Picture from Flickr user Anjuli: http://www.flickr.com/photos/49502989227@N01/56641591/

The tagging of tags is justified by the fact that a tag can have different meanings in different contexts: tagging the tags and the relations between them is used to disambiguate these contexts. For example the tag "tank" on the Flickr photo sharing service[2] is used to tag military vehicles[3], fish tanks[4] as well as a person[5]. Tag tagging allows a user to explain the meaning of her or his annotations. It also reveals the multiciplicity of meanings: by tagging "tank" with "fish" *and* "vehicle", the ambiguity becomes apparent and users can then decide to filter accordingly.

The operation of tagging introduces a relation which is not only functional (something has been tagged by somebody) but also a *meaningful* (for some reason, excluding spam, somebody tagged something with this particular tag). Indeed tagging a picture with "wheels" may relate to what the picture depicts, and a tag "travel" may relate to the origin of the picture. However, the meaning of the relation is not made explicit by the user at the moment of tagging: we believe that not having to think precisely to the relation and verbalise it as one would do in an ontology results in a smaller cognitive load for the user and is part of the appeal of tagging. In extreme tagging however, this relation itself can be tagged, later on, by any user. The operation of tagging relations between tags can naturally be expressed by triples, for example, if "_" represents the *implicit* relation introduced by the tagging operation itself, while "…" is used to represent *any tag*, relations can be: (resource, {_}, "wheels"), (resource, {_}, "travel"), (resource, {_}, ...), (resource, {"shows"}, "wheels"), (resource, {"represents"}, "wheels"), (resource, {...}, "wheels") or (resource, {"taken-during"}, "travel"), etc. (cf. Figure 2)

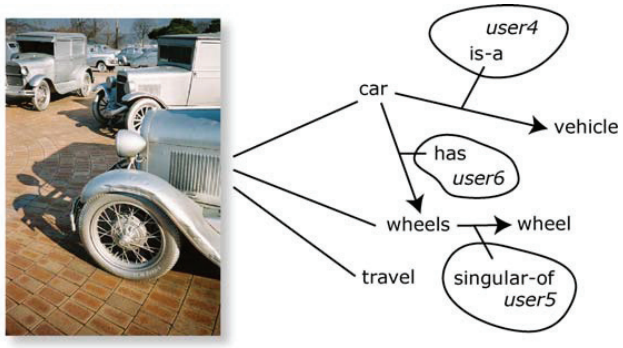


Figure 2. Tagging relations between tags.

Allowing users to tag the tags and the relations between them leads to the generation of *Semantic Associations*. Semantic Associations are chains of relations between one tag to another, or, in graph theoretic terms, a labelled path between two nodes. According to the definitions of [23] and [24], two entities are semantically associated

[2] Flickr, http://www.flickr.com/
[3] e.g. http://www.flickr.com/photos/barryslemmings/tags/tank/
[4] e.g. http://www.flickr.com/photos/towert7/tags/tank/
[5] http://www.flickr.com/photos/50836387@N00/tags/tank/

if they are *semantically connected*, i.e. there exist a path of relations between them, or *semantically similar*, i.e. two entities are similar if a path from the first one to another is similar to the path from the second one to another. We also call semantic annotations *knowledge paths*, as in this context they represent a crystallisation of the users' knowledge. We consider that the tagging relation itself, even if implicit, qualifies as a relation in a knowledge path, while we consider that the notion of semantic similarity can be extended from subclass/superclass relations only to any similarity measure. Collaboratively tagging resources, tags and relations leads to serendipitous discovery of associations between resources and/or tags. An example path between "wheel" and "vehicle" for example would be, expressed as a list of triples <"wheel", "vehicle"> = [("wheel", {"singular-of"}, "wheels"), ("wheels", {_}, "car"), ("car", {"is-a"}, "vehicle")].

The ETS model is defined as a collaborative tagging system with semantic associations. Therefore ETS are extensions of the formal model for collaborative tagging systems, defined as follows:

$$\Omega = \langle U, T, A, D \rangle \text{, where } A \subseteq U \times T \times T \text{ and } D \subseteq U \times T \times T \times T \quad . \tag{4}$$

We do not distinguish between the set of resources/entities *T* and the set of tags: all elements of *T* are entities, which can be "tags" or "resources". Indeed the mapping description of each entity by a unique identifier – in practice, a URI – makes the distinction superfluous. *A* is the set of assignments, as in traditional CTS while *D* represents directional annotations of relations between entities (tags or resources). According to this definition an ETS becomes a hypergraph:

$$\begin{aligned} &G = \langle V, E \rangle \text{, with vertices } \mathrm{V} = \mathrm{U} \cup \mathrm{T} \text{, and edges} \\ &E = \left\{ \{u, r, t, d\} \middle| (u, r, t) \in A \vee (u, r, t, d) \in D \right\} . \end{aligned} \tag{5}$$

The distinction between *A* and *D* reflects the distinction between *implicit* and *explicit* relations. An implicit relation occurs when an entity has been tagged while an explicit one appears if the relation between two entities has itself been tagged. A knowledge path is a path consisting of explicit or implicit relations between entities.

As relations between tags constitute triples, the link to RDF becomes obvious. Indeed ETS have the same goals as those sometime advocated by RDF proponents, "to allow anyone to say anything about anything" [25]. However, if ETS triples can be represented as RDF, extreme tagging introduces novelties. Indeed, RDF resources acquire their unique identity through the use of namespaces which contributes to slowing the process of knowledge acquisition as pre-existent knowledge about entities is needed. For example in the context of fish tanks the entity "http://fish.com/#tank" is needed, instead of "http://military.org/#tank". In ETS however, a tag is tagged by all its meanings, and disambiguation occurs during the query process, not at the tag description level, i.e. "tank" is only one tag, with a unique URI. If it is tagged as container and as a weapon, disambiguation will occur during knowledge path elicitation, as the knowledge path leading from "tank" to "fish" or "sea" will only use one of the meanings.

4 Tagopedia: an Extreme Tagging System

Tagopedia[6] is a prototype ETS built on top of the Facebook platform. Facebook is a social network web application providing a developer framework allowing the creation of applications which interact with core host features such as profile management and login. As any collaborative tagging system Tagopedia allows to tag resources, represented by URIs (cf. Figure 3).

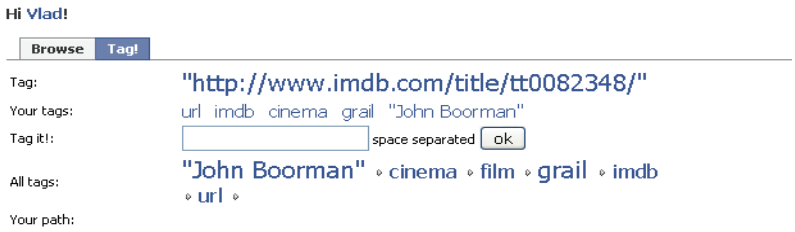


Figure 3. Basic Tagopedia usage.

When clicking on a tag however, the user is asked to tag the chosen relation or to enforce an already existing relation by selecting it (cf. Figure 4). The application then moves to the target tag, showing the linked tags and resources and allowing to define new relations. The user may also choose not to tag the relation and directly reach the target, keeping it implicit.

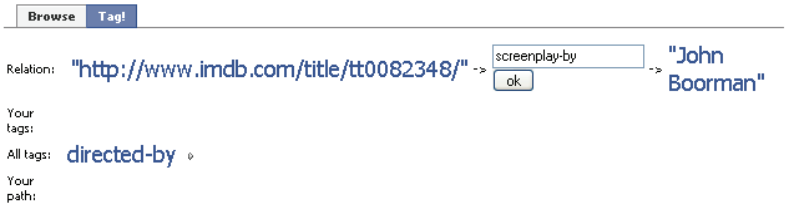


Figure 4. Tagging relations in Tagopedia.

Here is an example of collaboratively build semantic associations between entities in Tagopedia, written as a list of triples:

sa1:
[(“John Boorman”, {directed}, “Excalibur”),
(“Excalibur”, {about}, holy-grail) ,
(holy-grail, {similar-to}, grail),

[6] Available at http://apps.facebook.com/tagopedia/ (a Facebook account is required). The name *Tagopedia*, proposed independently by the authors, has already been proposed in 2005 by Russell Beattie in a blog post, for a related application (http://www.russellbeattie.com/notebook/1008277.html).

(grail, {topic-of}, “http://www.imdb.com/title/tt0071853/”),
(“http://www.imdb.com/title/tt0071853/”, {has-title}, “Monty Python and the Holy Grail”),
(“Monty Python and the Holy Grail”, {directed-by}, “Terry Gilliam”)]

sa2:
[(“John Boorman”, {is-a}, film-director), (film-director, {includes}, “Terry Gilliam”)]

sa3:
[(“John Boorman”, {is}, British), (British, {nationality-of}, “Terry Gilliam”)]

5 Emergent Semantics

In ETS, semantics are related to the users’ activity and input. The operations involved in a user’s activity can be classified as: *annotation*, *navigation* and *control*. At each level there is a need for *incitation*, a means to motivate the user to use the system. As a result of these three operations, tags are created and annotated collaboratively and unconstrained semantics emerge (cf. Figure 5). In this section, we describe each activity in turn as well as the corresponding motivation mechanism:

Figure 5. Plurality of meanings.

Through *annotation* users are given the opportunity to create their personal knowledge base. Indeed, instead of tagging resources at their hosting websites, building unrelated islands of tags, they can relate all their resources with their own meaning. Moreover, they can access, for the same resource, tags from other users, and decide to explore their meaning by navigating to them.

Through *navigation*, users build or enforce semantic associations. Indeed, by exploring a tag which tags a tag, either the user is looking for an explanation of this tag, or she already knows the relation. We assume that she knows the relation if a) she tagged it before, or b) she chooses to tag it when asked to do so. As previously mentioned, navigation does not happen between a tag and another tag without presenting the relation, which the user can choose to tag or not. The motivation of this additional step is to constrain the meanings obtained. Paths which have been explored and validated, are recorded and displayed the next time a request is made to find the paths from one node to another.

Finally *control* mechanisms are necessary in order for the system to evolve, some of these control mechanisms can be:

1) *total control over ones annotations*: the annotations added by a user can be modified or deleted by her.
2) *appreciation and depreciation of tags*: a user can rank a tag (+ or – only). If the total ranking goes below a given threshold, the tag becomes "private" and does not appear in public searches any more. A similar method is already used by commercial websites[7].
3) *questions to author*: Facebook, just as other social networks provides the notion of "friends", or "contacts", i.e. users which acknowledged a mutual relationship. If a user does not understand a tagging made by one of her related users a quick means is provided to send him or her a message to ask for an explanation, i.e. the tagging of this particular relation. The requester is notified as soon as the explanation has been given.

It is assumed that each user is interested in sharing his or her vision of the world and in discovering other ways of perceiving it. To the first interest corresponds the *annotation* activity as well as some control activities number 1) and 2). As a further incitation annotating increases the user's ranking, in a similar way as internet forums display titles according to the number of posts (often quite imaginative, for example using a graduated scale going from *rookie*, to *half-god* or *absolute guru*). In parallel, an increase in status can be achieved through *navigation* only, in a similar manner to some multiplayer computer games which increase the avatar's status by providing titles according to the percentage of the virtual map explored. Indeed, These two ways of using the system, annotating and creating, combine when a *navigator* – i.e. a user who mostly navigates the system, comparable to a Wikipedia reader rather than to an editor – earns creation points by completing paths and *creators* earn more points if the paths they created are navigated (i.e. if they make sense). Further incitation may involve *visualisation* of the number of elements created, as well as graph presentation of the paths explored.

6 Conclusion and Future Work

The benefits of pushing tagging to an extreme are the ease with which knowledge is acquired, as well as the comprehensiveness of the resulting KB. Possible caveats, which we believe can be solved by collaborative means, include the difficulty to assess the relevance of the resulting knowledge in a given context. Tagopedia is a first prototype of an Extreme Tagging System and we are waiting to obtain a larger knowledge base to attempt a serious evaluation. However, we used the prototype in a limited environment composed of 5 users, and, from the amount of serendipitous meaning collected, were already convinced of the interest of the system. We are planning to release it to the Facebook community in the following months and explore the

[7] e.g. Spockcom, http://www.spock.com/.

aforementioned control mechanisms through it. We are also working on an RDF export mechanism as well as on the integration of a SPARQL query engine. We also plan to import large amounts of tags from Wikipedia and other websites, using links inside the pages or other structured information in order to populate the knowledge base.

Acknowledgments. The authors are grateful to the anonymous reviewers for their precious comments. They would also like to thank Yiwen Wang, Kaixuan Wang and Fadi Badra, who contributed ideas and energy in the early stages of this project. Further thanks are extended to Sean Bechhofer, Enrico Motta, and John Domingue, who contributed, during SSSW07 and later on, to the success of this enterprise, and to Lyndon Nixon for proofreading this paper.

References

1. D. Fensel and M. Musen, "The semantic web: a brain for humankind", Intelligent Systems, vol. 16, pp. 24-25, 2001.
2. N. Guarino, Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy. IOS Press, 1998.
3. M. Dzbor, E. Motta, C. Buil, J. Gomez, O. Görlitz & H. Lewen: Developing ontologies in OWL: An observational study. OWL: Experiences & Directions Workshop, Georgia, US, 10-11 November 2006
4. M. d'Aquin, M. Sabou, M. Dzbor, C. Baldassarre, L. Gridinoc, S. Angeletou, and E. Motta. WATSON: A Gateway for the Semantic Web. Poster session of the European Semantic Web Conference, ESWC 2007.
5. The Semantic Web "Layer Cake": http://www.w3.org/2004/Talks/0412-RDF-functions/slide4-0.html
6. Sir Tim Berners-Lee at the Oxford Internet Institute, from webcast: http://webcast.oii.ox.ac.uk/?view=Webcast&ID=20060314_139
7. W. Kuhn: Why Information Science needs Cognitive Semantics - and what it has to offer in return, 2003.
8. A. Wierzbicka: "Apples" Are Not a "Kind of Fruit": The Semantics of Human Categorization American Ethnologist, JSTOR, 11, 313—328, 1984.
9. P. Gärdenfors: How to Make the Semantic Web More Semantic Formal Ontology in Information Systems, Proceedings of the Third International Conference (FOIS), 17-34, 2004.
10. C. Vangenot, C. Parent and S. Spaccapietra: Modeling and manipulating multiple representations of spatial data Proc. of the Symposium on Geospatial Theory, Processing and Applications, 2002.
11. J. Gibson: The Ecological Approach to Visual Perception, Lawrence Erlbaum Associates, 1979, p.42-43
12. A. Mathes: Folksonomies-Cooperative Classification and Communication Through Shared Metadata Computer Mediated Communication, LIS590CMC (Doctoral Seminar), Graduate School of Library and Information Science, University of Illinois Urbana-Champaign, December, 2004
13. S. Golder & B. Huberman: The Structure of Collaborative Tagging Systems, Arxiv preprint cs.DL/0508082, 2005
14. Angeletou, S., Sabou, M., Specia, L., Motta, E., (2007) Bridging the Gap Between Folksonomies and the Semantic Web: An Experience Report. Workshop: Bridging the Gap between Semantic Web and Web 2.0, European Semantic Web Conference.

15. Vander Wal, T. Folksonomy, http://www.vanderwal.net/folksonomy.html, 2007
16. Mika, P. Ontologies are us: A unified model of social networks and semantics Proc. ISWC2005, 2005
17. P. Schmitz. Inducing Ontology from Flickr Tags. In Proc. of the Collaborative Web Tagging Workshop at WWW'06, 2006.
18. X. Wu, L. Zhang, and Y. Yu. Exploring Social Annotations for the Semantic Web. In Proc. of WWW'06, 2006.
19. G. Begelman, P. Keller, and F. Smadja. Automated Tag Clustering: Improving search and exploration in the tag space. In Proc. of the Collaborative Web Tagging Workshop at WWW'06, 2006.
20. Gruber, T., Ontology of Folksonomy: A Mash-up of Apples and Oranges. AIS SIGSEMIS Bulletin, 2005. 2 (3&4).
21. Schmitz, C.; Hotho, A.; Jaschke, R. & Stumme, G. Mining association rules in folksonomies Data Science and Classification: Proc. of the 10th IFCS Conf., Ljubljana, Slovenia, July, Springer, 2006
22. L. Specia and E. Motta. Integrating Folksonomies with the Semantic Web. In Proc. of ESWC'07, 2007.
23. A. Sheth et al., Semantic Association Identification and Knowledge Discovery for National Security Applications. 2004.
24. K. Anyanwu and A. Sheth, "P-Queries: enabling querying for semantic associations on the semantic web", Proceedings of the 12th international conference on World Wide Web, pp. 690-699, 2003.
25. I. Davis, Introduction to RDF slides, http://research.talis.com/2005/rdf-intro/#(7), 2005

The HCOME-3O Framework for Supporting the Collaborative Engineering of Evolving Ontologies

George A. Vouros[1], Konstantinos Kotis[1], Christos Chalkiopoulos[1], Nikoleta Lelli[1]

[1] University of the Aegean, Dept. of Information & Communications Systems Engineering,
AI Lab,
83200 Karlovassi, Greece
{georgev, kotis}@aegean.gr
http://www.icsd.aegean.gr/ai-lab

Abstract. Nowadays it is widely accepted that ontologies, the key technology for the realization of the Semantic Web, are artefacts that are collaboratively and iteratively developed/evolved, shared, evaluated and discussed within communities of knowledge workers. To enhance the potential of ontologies to be collaboratively engineered and be consistently evolved within and between different communities, they must be escorted with rich meta-information describing the conceptualisations they realize, implementation decisions, the rationale for their evolution, as well as the evolution itself. To support the collaborative engineering of ontologies within and across different communities, this paper proposes a framework of (meta-)ontologies for capturing the meta-information that is necessary for interlinking, sharing, and combining knowledge among the parties involved in such a process. The framework is being embedded in the HCOME ontology engineering methodology, and can be applied to the design and implementation of ontology engineering tools towards advancing their interoperability.

1 Introduction

Ontologies establish a common vocabulary for community members to interlink, combine and communicate knowledge shaped through practice and interaction, binding the knowledge processes of creating, importing, capturing, retrieving, and using knowledge [11]. The ontology engineering process itself involves knowledge-intensive activities performed by members of specific communities. People participating in such a process need to share a common understanding of the various aspects and issues involved i.e. domain, methodological and tool-related ones. Therefore, (meta-)ontologies can play a major role in interlinking, sharing and combining information among the parties involved in a collaborative ontology engineering process.

We distinguish between domain knowledge and development information involved in the ontology engineering process. Domain knowledge concerns the conceptualization(s) that knowledge workers shape in order to develop a domain-specific ontology. Development information concerns a) the language-specific aspects for formalizing conceptualizations b) the interlinking of the conceptualizations with domain-related

resources and collaborating parties, c) the recording of developers' rationale on choosing specific conceptualizations and ways of formalizing them, and d) the ontology evolution i.e. the changes performed on (informal or formal) conceptualizations and the clustering of these changes in different versions of a domain ontology.

This paper focuses on the formal specification of development information in order to support advanced collaborative ontology engineering processes for the specification of continuously evolving domain knowledge.

Recent ontology engineering methodologies (HCOME [5], DILIGENT [10]) emphasize on (a) the incorporation of ontology engineering tasks in knowledge-empowered organizations in ways that are seamless to the day-to-day activities of the organization members and on (b) the active and decisive involvement of the knowledge workers in all stages of the ontology engineering processes. Particularly, the HCOME methodology accentuates the active and decisive participation of knowledge workers in the ontology life-cycle. Doing so, domain ontologies are developed and managed according to knowledge workers' abilities, they are developed individually as well as conversationally, and they are put in the context of workers' experiences and working settings, as an integrated part of workers' "knowing" process. Besides the methodological issues, leveraging the role of knowledge workers in the ontology life-cycle entails the development of ontology engineering tools that provide greater opportunities for them to manage and interact with their conceptualizations in a direct and continuous way, not only by reusing and combining domain/development knowledge but also communicating such knowledge between them effectively.

This paper points that to empower knowledge workers to actively and decisively participate in the ontology life-cycle, we need to establish a common understanding of (or at least make explicit to them) the way(s) that ontologies are being implemented and evolved. Towards this target, this paper proposes a framework of (meta-)ontologies for capturing the development information that is necessary for interlinking, sharing and combining knowledge among the parties involved in a collaborative ontology engineering process. This framework is embedded in the HCOME methodology, advancing the potential for collaborative ontology engineering tasks, and the interoperability of ontology engineering tools by applying it to their design and implementation.

This paper is structured as follows: Section 2 provides the motivation and the work that is closely related to the aims of our work and section 3 presents the proposed framework. Section 4 presents preliminary evaluation of the framework using a collaborative ontology engineering tool, showing its potential to satisfy the stated requirements.

2 Motivation and Related Work

Knowledge workers within and across communities, even if they are interested in the same domain, may not share the same context. The context includes the background knowledge that community members have, their commonly accepted practices, their experiences concerning the domain of interest, their interests and motivation to exploiting ontologies, as well as the ontology exploitation tools/applications they use.

More important to the exploitation and evolution of living ontologies, communities may not have the same view of how and why domain ontologies have been developed and/or evolved in the way they did, and they may not even use the same tool or methodology to engineer them. Therefore, (meta-)ontologies, besides facilitating a common understanding of the issues involved in the ontology engineering task (which is essential for people working with different ontology engineering methodologies to communicate), they also provide a common vocabulary for sharing information concerning the development of domain ontologies (which is essential for different ontology engineering tools to interoperate), and specific information concerning their evolution (which is essential for people to inspect and assess the changes made to domain ontologies).

(Meta-)ontologies must support the sharing, reuse and consistent evolution of domain ontologies within and across communities. This implies the need for the extended sharing of the constructed domain ontologies, together with *formal* specification of meta-information that would support the interlinking, combination, and communication of knowledge shaped through practice and interaction among community members.

Ontologies for the specification of such meta-information must support:

1. The identification of those parties that contribute to the development/evolution of a single ontology.
2. The recording of the conversations towards the commonly agreed requirements and scope of the ontology.
3. The tracking of the arguments towards the agreed (formal or informal) specifications
4. Tracking the change operations performed by individual users
5. Capturing the informal meaning of ontology elements by interlinking formal specifications to other domain resources (e.g. thesaurus, lexicons).
6. The specification of change operations that have occurred between two subsequent ontology versions.
7. Structured argumentation dialogues for the evaluation and further development/evolution of shared ontologies.
8. Integration of versioning and change-tracking information with argumentation dialogues, for the effective sharing of ontologies: This enables tracking the rationale behind individual changes, ontology versions, specification and implementation decisions.
9. The inter-contextual sharing of domain ontologies: Although previous work has emphasized on the sharing of ontologies within specific contexts, meta-information must support the inter-contextual sharing of ontologies, capturing all the detailed aspects involved in the development/evolution of ontologies, either in a personal or in a shared space.

The above requirements for meta-information point to the need of an integrated framework of (meta-)ontologies for the intertwined specification of (a) structured argumentation dialogues, (b) change operations and ontology versions during ontology evolution, (c) administrative information concerning domain conceptualization and ontologies implementations, contributors involved in ontology lifecycle, and relations to other domain-related resources.

Viewing this framework in the context of a specific collaborative ontology engineering methodology, it aims to advance the potential of reusing and consistently evolving formal conceptualizations of domain knowledge. We view this as an essential requirement to the use of such a framework of meta-information as it assures that the framework facilitates the ontology engineering process and advances the understanding of methodological issues involved in the engineering of ontologies (i.e. what does a collaborative ontology engineering process involves, who may participate and what is expected/permitted to contribute, what changes are expected to be made, how versions are being assimilated, the degree to which these changes/versions have to be justified). Specifically, we aim to advance the HCOME methodology by incorporating a framework of (meta-)ontologies to support the collaborative ontology engineering process. HCOME has accentuated the need for advanced functionality for engineering shared and continuously evolving ontologies. HCOME places major emphasis to the conversational development, evaluation and evolution of ontologies. This implies the need for the extended sharing of the constructed domain ontologies, together with meta-information that supports the interlinking, combination, and communication of knowledge shaped through practice and interaction among community members, binding the knowledge processes of creating, importing, capturing, retrieving, and using knowledge.

In the current bibliography, there are works about (meta-)ontologies for supporting specific facets of the ontology engineering process (OMV [1, 2], OntoView [3], CHAO [9], DILIGENT [12]): These works do not provide an integrated solution to the problem of knowledge exchange, so as to support the collaborative engineering and consistent evolution of ontologies within and across different communities of knowledge workers. Although they do specify types of information that need to be captured for describing the conceptualization and development of domain ontologies, they do not specify a unique integrated conceptual framework for capturing and sharing this information, and neither specify how such a framework is embedded within an ontology engineering methodology.

Related work concerning ontology evolution frameworks in specific, has been proposed in [9], using the Change and Annotation Ontology (CHAO). Instances of this ontology represent changes between two versions of an ontology. Changes are linked to annotations. For each change, the change and annotation ontology describes the following information: the type of change; the class, property, or instance that was changed; the user who performed the change; the date and time when the change was performed. Although annotations on changes are being recorded, the arguments supporting and/or being against individual changes are not captured, affecting the effectiveness of the representation for recording the rationale and different views/opinions behind individual changes and/or the issuing of assimilated ontology versions.

Other works [6, 7, and 8] provide information concerning ontology change management in different levels of abstractions (simple or complex changes, collections of changes (versions), changes discovered from similarity measures, etc). However, although annotations on changes are being recorded, arguments are not captured and are not interrelated with other meta-information.

Similarly to the ontology-evolution framework proposed in [9], Figure 1 presents the processes that may be performed by knowledge workers and the meta-information

that must be recorded as a by-product of the collaborative ontology engineering processes according to HCOME: As it can be seen, in extend to other frameworks (e.g. in [9]), we require ontologies to be escorted with the meta-information concerning their development and evolution. This meta-information is further enriched via the processes of editing (creating, importing, capturing), exploiting (inspecting, retrieving and using) and arguing about domain knowledge. We further require that when domain ontologies or parts of them are being shared between workers, the relevant meta-information has to be shared as well.

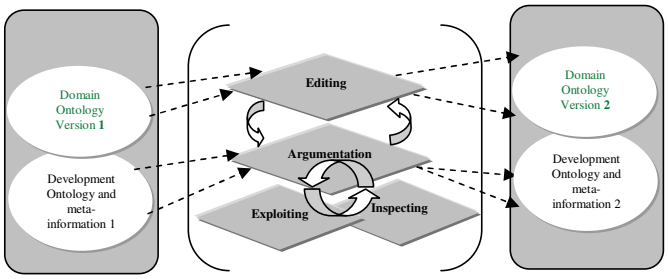


Fig. 1. Processes and meta-information in an ontology evolution cycle: Rectangles denote processes and ovals ontologies. Plain arrows point on the input and output produced: domain ontologies and individuals recorded in (meta-)ontologies.

Summarizing the above, the proposed work aims to advance the state of the art by contributing to the following issues conjunctively:

1. it provides an integrated framework of ontologies for the specification of meta-information,
2. it embeds this framework within the HCOME collaborative methodology for ontology engineering,
3. it examines the implications of adopting this framework to the design of ontology engineering tools.

3 The HCOME-3O framework

According to the stated requirements, this section presents the HCOME-3O framework of three ontologies, which specify meta-information concerning:

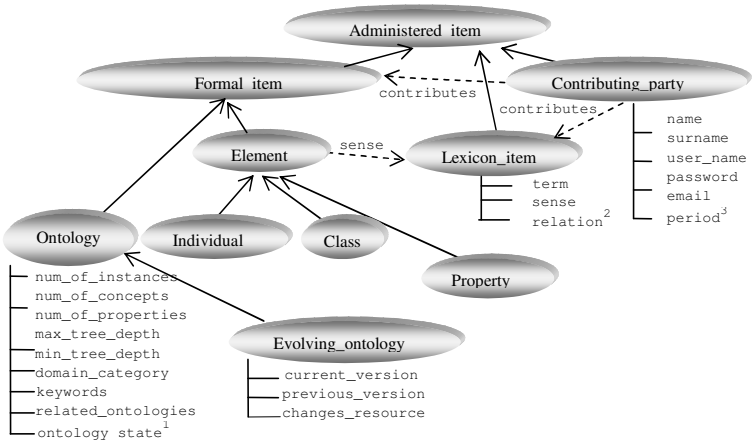
Administrative meta-information: This specifies information concerning the conceptualization, development of domain ontologies, as well as versioning of ontologies.

Change operations meta-information: This concerns changes that have been made in domain ontologies.

Argumentation meta-information: It concerns rationale and arguments related to individual changes and ontology versions.

Although ontologies such as the ones presented in section 3 could be incorporated into the framework, we have only consulted these ontologies in the engineering of the proposed framework, so as to specify the minimum meta-information that must be captured in a modular but intertwined manner, according on the stated requirements.

3.1 Administrative meta-information



1. It specifies whether an ontology is personal, shared or agreed
2. It specifies the semantic relation between the term that lexicalizes the concept and the lexicon/thesaurus term entry (e.g. synonym, more specific, more general)
3. A period starts when a personal ontology is send to the shared space and ends when a version of this ontology is in the agreed state.

Fig. 2. Administrative meta-information

The Administration ontology provides a schema for representing meta-information about administered items and contributing parties. `Administered items` can be either ontologies, ontology elements (classes, properties, individuals), or items that informally describe the meaning of terms that lexicalize properties or classes in the domain ontology. All types of items are identified by a resource identifier. `Formal items` and `lexicon items` are contributed by `contributing parties`. Lexicon items may also be automatically assigned by mapping algorithms. Contributing parties may contribute to the development/evolution of a personal, shared or agreed `ontology`, or may contribute to the specification of a `class`, `property` or `individual`. Also, an ontology can have several uniquely identified versions, which result from the changes made and recorded during ontology development/evolution.

The administrative ontology distinguishes between the informal and formal conceptualization of a domain by linking items to the informal (lexicon-based) description of their meaning: This distinction is further supported by linking items to argumentation items (of the argumentation dialogue) that provide arguments for the conceptualizations/specifications made. In this way, administrative meta-information is documentary and extensible, and supports the interlinking with other domain-specific resources. The entities of the implemented proposed schema and their relations are depicted in Figure 2.

3.2 Change operations meta-information

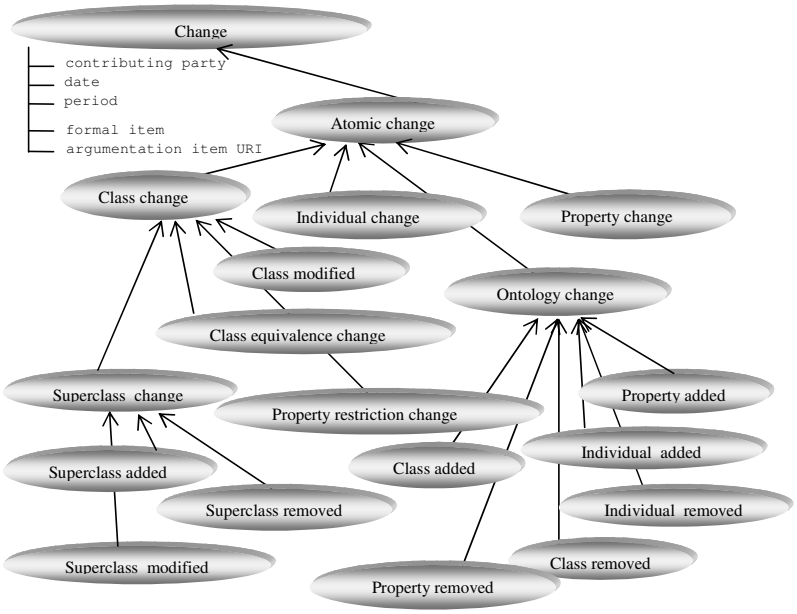


Fig. 3. Meta-information concerning changes that occur during the editing of ontologies

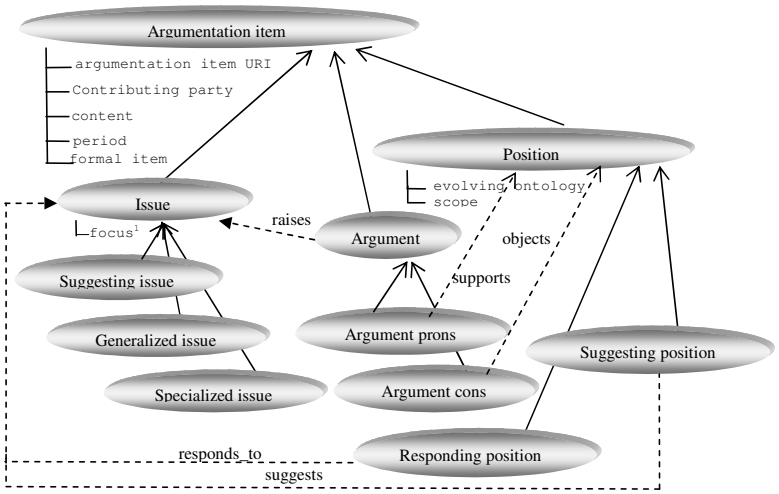
The change operations (meta-)ontology provides a schema for representing information about the `changes` that contributing parties can make to the ontology elements during the evolution of a domain ontology. It also supports the reporting of differences between two versions of a single ontology.

This ontology currently specifies only atomic changes: Any atomic change to the specification of a formal element (Class, Property, and Individual) made during the editing of an ontology is recorded together with the rationale behind it. The relations between a change made by a contributed party, the argumentation items (if any) be-

hind this change, and the element that has been changed, are specified by means of the `Atomic change` class properties (contributing party, argumentation item, formal item).

Figure 3 depicts only a part of the ontology. Change operations that can apply to individuals and properties are missing due to space restrictions.

3.3 Argumentation meta-information



1. It specifies whether the issue concerns the conceptualization of the domain or formal aspects of ontology implementation
2. It specifies whether the position concerns the ontology version or the change of an ontology element

Fig. 4. Information captured in an argumentation dialogues.

The argumentation ontology provides a schema for representing meta-information about *issues*, *positions*, and *arguments* that contributing parties make during an argumentation dialogue upon the collaborative evolution of shared ontologies.

Specifically, an argument may raise an issue that either suggests changes in the domain conceptualization, or questions the implementation of the conceptualized entities/properties. Based on this issue, a collaborative party may respond by publicizing a position, i.e. a new version of the ontology, or by suggesting the change of a specific ontology element. A new argument may be placed for or against a position, and so on. Issues may be generalized or specialized by other issues. The connection of the recorded arguments with the ontology elements discussed by specific contributing parties and with the changes made during a period (Figure 3) is performed through the

argumentation item and position classes' properties (formal item, contributing party, period, evolving ontology).

The argumentation ontology supports the capturing of the structure of the entire argumentation dialogue as it evolves among collaborating parties within a period. It allows the tracking and the rationale behind atomic changes and/or ontology versions. It is generic and simple enough so as to support argumentation on the conceptual and on the formal aspects of an ontology.

The entities of the implemented proposed schema and their relations are depicted in Figure 4.

4 Preliminary Evaluation

Early evaluation of the proposed framework has been performed by embedding it in a prototype version of HCONE tool [4]. This version was designed by taking into account the requirements of the proposed framework in addition to the methodological requirements of HCOME methodology. Having said that, it must be clearly stated that in this paper we do not point on the value of a collaborative engineering methodology itself. The contribution and importance of collaborative engineering of ontologies has been studied in other related works [5, 10] and evaluated in [13, 14].

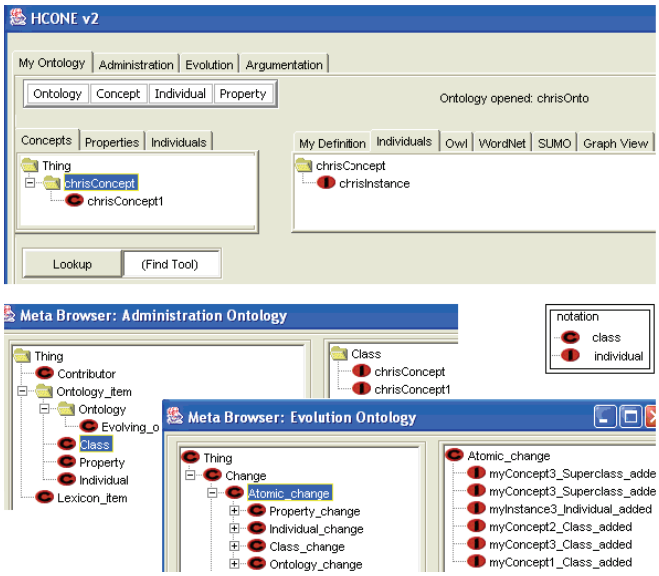


Fig. 5. The HCONE GUI and the meta-browsers windows for exploring the recorded meta-information

An early snapshot of the new HCONE tool is depicted in Figure 5, showing the supported functionalities. In the same figure, the meta-browser windows for navigating through the recorded administrative and evolution meta-information (populated meta-ontologies) are also depicted.

Embedding the proposed framework in HCONE tool allows the recording and presentation of ontologies' development information. This information is recorded as instances of OWL[1]-implemented (meta-)ontologies and is stored in a triples-like RDF[2] store (JENA[3]). The value of the framework in the engineering of shared and evolving ontologies can be measured by the capability of the proposed-framework-based tool to record and present ontologies' development information. Figure 5 shows a snapshot of the recorded meta-information, as it can be explored using the HCONE tool ontology-browser tools.

Prototype implementation has demonstrated that the proposed framework is capable of recording and presenting the following meta-information:

1. Meta-information concerning the parties that contributes to the development/evolution of a single ontology. Such information is recorded as individuals of class "Contributor" in the Administration (meta-)ontology. "Contributor" individuals are related to ontology development and their evolution ("Formal Item" class of Administation meta-ontology) through "contributes" property. Individuals are returned from queries executed over the Administration (meta-)ontology e.g. "*Find all contributors (individuals of "Contributor" class) which contribute to ("contributes" property) the development of "myOntology" ontology (individual of Ontology class)* ".
2. Meta-information concerning the recording and tracking of the conversations. Such information is recorded as individuals of "Argumentation Item" class (specifically, individuals of its subclasses) of the Argumentation (meta-)ontology. "Argumentation Item" individuals are related to a specific ontology development ("Formal Item" class of Administation meta-ontology) through their "formal item" property in the Argumentation (meta-)ontology. Individuals are returned from queries executed over the Argumentation (meta-)ontology e.g. "*Find all the "Argumentation items" (individuals of all subclasses of "Argumentation item" class) which are related to a specific ontology element ("Formal item" property)*".
3. Meta-information concerning the recording of the interlinking between conversations and ontology evolution (versions of a domain ontology). Such information is recorded as a value of the "evolving ontology" property of the "Position" class of the Argumentation (meta-)ontology. This value represents the ontology version of a domain ontology that a contributor has developed in his personal space, and which is uploaded in the shared space for inspection by other contributors.
4. Meta-information concerning the tracking of change operations performed on specific ontology items by individual users. Such information is recorded as individuals of class "Change" (specifically, individuals of its subclasses) of the Evolution

[1] http://www.w3.org/TR/owl-ref/
[2] www.w3.org/**RDF**/
[3] http://jena.sourceforge.net/

(meta-)ontology. Individuals of Class "Change" are related to a specific ontology development ("Formal Item" class of Administration (meta-)ontology) through their "Formal item" property in the Evolution (meta-)ontology. Individuals are returned from queries executed over the Evolution (meta-)ontology "*Find all the changes (individuals of "Change" class subclasses) which are related to a specific domain ontology element ("Formal item" property)*".

5. Meta-information concerning the integration of versioning and change-tracking information with argumentation dialogues. Further, to enable tracking of the rationale behind individual changes, ontology versions, specification and implementation decisions, meta-information concerning the discussions upon specific ontology elements is recorded. Individuals of class "Change" are related to a specific period of discussions upon a specific ontology element through their "period" property of the Evolution (meta-)ontology. The correspondent value of "period" property of the Evolution (meta-)ontology is currently obtained in a rather mediated manner: it is obtained from the argumentation ontology and represents the period that a specific ontology element has been discussed (i.e. related to a specific argumentation item) and this ontology element has been involved into a change operation. Thus, a specific individual change will not be interlinked to an argumentation dialogue unless the ontology element that this change concerns is interlinked to an argumentation item.
6. Meta-information concerning the capture of all the detailed aspects involved in the development/evolution of ontologies, either in a personal or in a shared space. When a knowledge worker fetches an ontology version from the central ontology store, she/he gets all the related meta-information: the previous version, the change operations, the argumentation items related to these versions, administrative meta-information. This allows him/her to inspect the evolution history and decide on the exact contributions he/she has to make. To meet this requirement we have designed and implemented a central ontology repository which stores both domain and meta-information ontologies in a triple-like RDF store (Relational database). When a domain ontology version is accessed using the HCONE environment, several queries are executed against all the stored information in the database in order to load individual values of meta-ontologies concerning this particular ontology. The linking property between all the related information of a specific domain ontology version that is retrieved by these queries is the "Formal item" property which represents a unique identifier (URI) for a specific ontology or ontology element resource. A domain ontology is personal i.e. only one contributor can manage it (its creator) until it is uploaded to an argumentation dialogue for discussion. In this case the property "ontology state" of class "Ontology" in the Administration (meta-)ontology takes the value "shared". If all contributors that have joined the specific dialogue "agree" on the shared ontology, the "ontology state" property is assigned the value "agreed". An "agreed" or "shared" ontology is accessible and manageable by all its contributors. HCONE utilizes the Administration meta-information in order to manage contributors' rights on accessing domain ontologies.

The preliminary evaluation of the proposed framework embedded in HCONE tool has been conducted with test ontologies in an experimental networked setting of a small group of collaborating users. Important issues such as scalability and usability of the

prototype tool have been taken into account during tool design. For instance, persistent storage technology at the server-side has been used for handling the possibility of large scale ontologies. A large scale evaluation of the HCONE tool with real-world collaborative ontology engineering tasks has been planned in the near future.

5 Conclusions

To enhance the potential of evolving ontologies to be collaboratively engineered within and between different communities, we have proposed an integrated framework of three (meta-)ontologies that provide information concerning the conceptualization and the development of domain ontologies, atomic changes made by knowledge workers, long-term evolutions and argumentations behind decisions taken during the lifecycle of an ontology. This framework has been proposed in the context of HCOME collaborative engineering methodology and suggested for advancing the functionality of ontology-engineering tools, pointing to specific design issues.

Further work concerns the implementation of further advanced functionalities in the HCONE tool that will also uncover new implications as far as the HCOME-3O framework potential is concerned. More specific, meta-information that is not yet recorded and presented through the HCONE implementation concerns the capturing of the informal meaning of ontology elements by interlinking formal specifications to other domain resources (e.g. thesaurus, lexicons). Also, we must provide a more sophisticated mechanism for interlinking individual changes of ontology elements with specific argumentation items of a discussion period that have actually suggested and influence a change, not with the whole discussions and certainly not with items that have been related with a change for some reason but eventually they did not influence the change at all. Finally, we could advance the changes operations and extend the ontology to represent more complex changes i.e. composite changes that influence more than one ontology element (atomic change).

References

1. Hartmann J., Sure Y., Haase P., Palma R., Suárez-Figueroa M.: OMV -- Ontology Metadata Vocabulary. In Chris Welty, ISWC 2005 - In Ontology Patterns for the Semantic Web (2005)
2. Hartmann J., Sure Y., Haase P., Suárez-Figueroa M., Studer R., Gómez-Pérez A., Palma R.: Ontology Metadata Vocabulary and Applications. In Robert Meersman, Zahir Tari, Pilar Herrero et al., International Conference on Ontologies, Databases and Applications of Semantics. In Workshop on Web Semantics (SWWS), Springer (2005) pp. 906-915
3. Klein M., Fensel D., Kiryakov A., Ognyanov D.: Ontology Versioning and Change Detection on the Web. EKAW 2002, 197-212
4. Kotis K. and Vouros G.: Human Centered Ontology Management with HCONE. IJCAI'03, Ontologies and Distributed Systems Workshop, Acapulco, Mexico (2003) CEUR-WS.org/Vol. 71

5. Kotis K. and Vouros G.: Human-Centered Ontology Engineering: the HCOME Methodology. International Journal of Knowledge and Information Systems (KAIS), (Published Online First: 9 Sept. 2005) Springer (2006) 10(1):109-131
6. Liang Y., Alani H., Shadbolt N. R.: Change Management: The Core Task of ontology Versioning and Evolution. In Proceedings of Postgraduate Research Conference in Electronics, Photonics, Communications and Networks, and Computing Science 2005 Lancaster, United Kingdom. (PREP 2005), pp. 221-222
7. Maedche A., Motik B., Stojanovic L., Studer R., Volz R.: Managing Multiple Ontologies and Ontology Evolution in Ontologging. In Proceedings of the Conference on Intelligent Information Processing, World Computer Congress 2002, Montreal, Canada. Kluwer Academic Publishers (2002)
8. Noy N. F., Klein M.: Tracking Complex Changes During Ontology Evolution. Third International Conference on the Semantic Web (ISWC-2004), Hiroshima, Japan
9. Noy N. F., Chugh A., Liu W., Musen M.: A Framework for Ontology Evolution in Collaborative Environments. 5th International Semantic Web Conference, Athens, GA, (2006)
10. Pinto, H. S., Staab, S., Tempich, C.: DILIGENT: Towards a fine-grained methodology for Distributed, Loosely-controlled and evolving Engineering of oNTologies. ECAI 2004: 393-397
11. Staab S, Studer R, Schnurr H, Sure Y.: Knowledge Processes and Ontologies. IEEE Intelligent Systems, (2001) pp 26-34
12. Tempich C., Pinto H. S., Sure, Y., Staab, S.: An Argumentation Ontology for DIstributed, Loosely-controlled and evolvInG Engineering processes of oNTologies (DILIGENT) In Asunción Gómez-Pérez and Jérôme Euzenat, Second European Semantic Web Conference, (ESWC 2005), volume 3532 of LNCS, pp. 241--256. Springer, Heraklion, Crete, Greece, May 2005.
13. Tempich C., Pinto H. S., Sure, Y., Vrandecic, D., Casellas, N., Casanovas, P.: Evaluating DILIGENT Ontology Engineering in a Legal Case Study In Pompeu Casanovas, Pablo Noriega, Daniele Bourcier, V.R.Benjamins, IVR 22nd World Congress - Law and Justice in a Global Society, no. B4, pp. 330--331.
14. Tempich C., Pinto H. S., Staab S.: Ontology Engineering Revisited: An Iterative Case Study. ESWC 2006: 110-124

Understanding the Semantics of Ambiguous Tags in Folksonomies

Ching-man Au Yeung, Nicholas Gibbins, and Nigel Shadbolt

Intelligence, Agents and Multimedia Group (IAM),
School of Electronics and Computer Science,
University of Southampton,
Southampton SO17 1BJ, UK
{cmay06r,nmg,nrs}@ecs.soton.ac.uk

Abstract. The use of tags to describe Web resources in a collaborative manner has experienced rising popularity among Web users in recent years. The product of such activity is given the name folksonomy, which can be considered as a scheme of organizing information in the users' own way. In this paper, we present a possible way to analyze the tripartite graphs – graphs involving users, tags and resources – of folksonomies and discuss how these elements acquire their meanings through their associations with other elements, a process we call mutual contextualization. In particular, we demonstrate how different meanings of ambiguous tags can be discovered through such analysis of the tripartite graph by studying the tag *sf*. We also discuss how the result can be used as a basis to better understand the nature of folksonomies.

1 Introduction

The use of freely-chosen words or phrases called tags to classify Web resources has experienced rising popularity among Web users in recent years. Through the use of tags, Web users come to share and organize their favourite Web resources in different social tagging systems, such as del.icio.us[1] and Flickr[2]. The result of this collaborative and social tagging activity is given the name folksonomy, which refers to the classification system evolved from the individual contributions of tags from the users [1].

Collaborative tagging possesses a number of advantages which account for its popularity. These include its simplicity as well as the freedom enjoyed by the users to choose their own tags. However, some limitations and shortcomings, such as the problem of ambiguous meanings of tags and the existence of synonyms, also affect its effectiveness to organize resources on the Web. As collaborative tagging attracts the attentions of researchers, methods on how useful information can be discovered from the seemingly chaotic folksonomies have been developed. In particular, some focus on discovering similar documents or communities of

[1] http://del.icio.us/
[2] http://www.flickr.com/

shared interests [17, 13], while some perform analysis on the affiliation between entities to find out different relations between tags [10, 14].

In this paper we focus on analysis of tripartite graphs of folksonomies, graphs which involve the three basic elements of collaborative tagging, namely users, tags and resources. We present how these elements come to acquire their own semantics through their connections with other elements in the graphs, a process which we call mutual contextualization. In particular, we carry out a preliminary study on tripartite graphs with data obtained from del.icio.us, and demonstrate how we can understand the semantics of ambiguous tags by examining the structures of these graphs. We also discuss how the result can be used as a basis to acquire a better understanding of the nature of folksonomies.

The rest of this paper is structured as follows. Section 2 gives some background information on collaborative tagging systems and folksonomies. We describe the process of mutual contextualization between the three basic elements in Section 3. We detail the preliminary study on tripartite graphs of folksonomies in Section 4, followed by discussions in Section 5. Finally we present our conclusions and discuss possible future research directions in Section 6.

2 Background

2.1 Collaborative Tagging Systems

Tagging originates from the idea of using keywords to describe and classify resources. These keywords are descriptive terms which indicate the topics addressed by the resources. Collaborative tagging systems emerged in recent years have taken this idea further by allowing general users to assign tags, which are freely-chosen keywords, to resources on the Web. For example, one can store a bookmark of the page "http://www.google.com/" on a collaborative tagging system, and assign to it the tags *google*, *search* and *useful*. As the tags of different users are aggregated, the tags form a kind of signature of the document, which can be used for future retrieval or indication of the nature of the page.

Collaborative tagging systems have started to thrive and grow in number since late 2003 and early 2004 [6]. As one of the earliest initiative of collaborative tagging, del.icio.us provides a kind of social bookmarking service, which allows users to store their bookmarks on the Web, and use tags to describe them. Other services focusing on different forms of Web resources appeared shortly. For example, Flickr allows users to tag digital photos uploaded by themselves.

Collaborative tagging are generally considered to have a number of advantages over traditional methods of organizing information, as evidently shown by its popularity among general Web users and its application on a wide range of Web resources. The following features of collaborative tagging are generally attributed to their success and popularity [1, 15, 18].

Low cognitive cost and entry barriers The simplicity of tagging allows any Web user to classify their favourite Web resources by using keywords that are not constrained by predefined vocabularies.

Immediate feedback and communication Tag suggestions in collaborative tagging systems provide mechanisms for users to communicate implicitly with each other through tag suggestions to describe resources on the Web.

Quick Adaptation to Changes in Vocabulary The freedom provided by tagging allows fast response to changes in the use of language and the emergency of new words. Terms like *AJAX*, *Web2.0*, *ontologies* and *social network* can be used readily by the users without the need to modify any pre-defined schemes.

Individual needs and formation of organization Tagging systems provide a convenient means for Web users to organize their favourite Web resources. Besides, as the systems develop, users are able to discover other people who are also interested in similar items.

On the other hand, limitations and problems of existing collaborative tagging systems have also been identified [1, 13, 18]. These issues hinder the growth or affect the usefulness of the systems.

Tag Ambiguity Since vocabulary is uncontrolled in collaborative tagging systems, there is no way to make sure that a tag is corresponding to a single and well-defined concept. For an example, items being tagged by the term *sf* may either be related to something about science fiction or the city San Francisco.

The use of multiple words and spaces Some systems allow users to input tags separated by spaces. Problems arise when users would like to use phrases with multiple words to describe the Web resources.

The problem of synonyms Different tags can be used to refer to the same concept in a tagging system. For example, "mac," "macintosh," and "apple" can all be used to describe Web resources related to Apple Macintosh computers[1]. The use of different word forms such as plurals and parts of speech also exacerbate the problem.

Lack of semantics A tag provides limited information about the documents being tagged. For example, when tagging an URL with the tag "podcast," one can mean that the website provides podcast, describes the use of podcast, or provides details on the history of podcasting.

2.2 Folksonomies

As more tags are contributed to a collaborative tagging system by the users, a form of classification scheme will take shape. Such scheme emerges from the collective efforts of the participating users, reflecting their own viewpoints on how the shared resources on the Web should be described using various tags. This product of collaborative tagging is now commonly referred to as *folksonomy* [16]. A folksonomy is generally agreed to be consisting of at least the following three sets of entities [9, 10, 18].

Users Users are the ones who assign tags to Web resources in social tagging systems. They are also referred to as actors, as in social network analysis.

Tags Tags are keywords chosen by users to describe and categorize resources. Depending on systems, tags can be a single word, a phase or a combination of symbols and alphabets. Tags are referred to as concepts in some works [10].

Resources Resources refer to the objects that are being tagged by the users in the social tagging systems. Depending on the system, resources can be used to refer to Web pages (bookmarks) as in del.icio.us or photos as in Flickr. Resources are also referred to as instances, objects or documents, depending on the context.

Quite a number of research works perform analysis on social tagging systems. However, even though most works adopt a model involving the above three entities, with a few mentioning extra dimensions such as the time of tagging, there is actually not a common consensus on the formal definition of folksonomy. Below we summarize the attempts in this respect.

Mika [10] represents a social tagging system as a tripartite graph, in which the set of vertices can be partitioned into three disjoint sets A, C and I, corresponding to the set of actors, the set of concepts and the set of objects being tagged. A folksonomy is then defined by a set of annotations $T \subseteq A \times C \times I$, an element of which is a triple representing an actor assigning a concept to an object being tagged.

Gruber [5] proposes a "tag ontology" which formalizes the activity of tagging through the use of an ontology. He suggests that tagging can be defined using a five-place relation: $Tagging(object, tag, tagger, source, [+/-])$, with object being the Web resources being tagged, tagger being the user who assigns tags, source being the system from which this annotation originates, and $[+/-]$ representing either a positive or negative vote placed on this annotation by the tagger. Newman [12] also developed a similar ontology for tagging. The act of tagging is modelled as a relation $T(Resource, Tagging(Tag, Agent, Time))$.

Hotho et al. [7] define a folksonomy as a tuple $\mathbb{F} := (U, T, R, Y, \prec)$. The finite sets U, T and R correspond to the set of users, tags and resources respectively. Y refers to the tag assignments, which are ternary relation between the above three sets: $Y \subseteq U \times T \times R$. $\prec$ is a user-specific relation which defines the sub/superordinate relations between tags. By dropping $\prec$, the folksonomy can be reduced to a tripartite graph, which is equivalent to Mika's model.

3 Mutual Contextualization in Folksonomies

The power of folksonomies lies in the interrelations between the three elements. A tag is only a symbol if it is not assigned to some Web resources. A tag is also ambiguous without a user's own interpretation of its meaning. Similarly, a user, though identified by its username, is characterized by the tags it uses and the resources it tags. Finally, a document is given semantics because tags act as a

form of metadata annotation. Hence, it is obvious that each of these elements in a folksonomy would be meaningless, or at least ambiguous in meaning, if they are considered independently. In other words, the semantics of one element depends on the context given by the other two, or all, elements that are related to it.

To further understand this kind of mutual contextualization, we examine each of the three elements in a folksonomy in detail. For more specific discussions, we assume that the Web resources involved are all Web documents. In addition, we define the data in a social tagging system, a folksonomy, as follows.

Definition 1. *A folksonomy F is a tuple $F = (U, T, D, A)$, where U is a set of users, T is a set of tags, D is a set of Web documents, and $A \subseteq U \times T \times D$ is a set of annotations.*

By adopting this definition, we are actually using the model described by Mika [10]. Since we are mainly focusing on the associations between the three elements and are obtaining data from a single social bookmarking site, information such as the time stamps and sources of tagging is irrelevant here. Thus, the definition we used here is a simple but sufficient one for our work presented here.

As we have mentioned, the three elements forming the tripartite graph of a social tagging system are users, tags and documents (resources). The tripartite graph can be reduced into a bipartite graph if, for example, we focus on a particular tag and extract only the users and documents associated with it. Since there are three types of elements, there can be three different types of bipartite graphs. This step is similar to the method introduced by Mika [10]. However, we distinguish our method from that presented by Mika by focusing on only one instance of a type (e.g. tags), instead of all the items of the same type, allowing us to acquire more specific understanding of the semantics of the instance.

3.1 Users

By focusing on a single user u, we obtain a bipartite graph TD_u defined as follows:

$$TD_u = \langle T \cup D, E_{td} \rangle, E_{td} = \{\{t, d\} | (u, t, d) \in A\}$$

In other words, an edge exists between a tag and a document if the user has assigned the tag to the document. The graph can be represented in matrix form, which we denote as $\mathbf{X} = \{x_{ij}\}$, $x_{ij} = 1$ if there is an edge connecting t_i and d_j. The bipartite graph represented by the matrix can be folded into two one-mode networks [10]. We denote one of them as $\mathbf{P} = \mathbf{X}\mathbf{X}'$, and another as $\mathbf{R} = \mathbf{X}'\mathbf{X}$.

$\mathbf{P}$ represents a kind of semantic network which shows the associations between different tags. It should be note that this is unlike the lightweight ontology mentioned in [10], as it only involves tags used by a single user. In other words, this is the personal vocabulary, a personomy [7], of a particular user.

The matrix $\mathbf{R}$ represents the personal repository of the user. Links between documents are weighted by the number of tags that have been assigned to both documents. Thus, documents having higher weights on the links between them are those that are considered by the particular user as more related.

3.2 Tags

By using a similar method as described above, we can obtain a bipartite graph UD_t regarding to a particular tag t:

$$UD_t = \langle U \cup D, E_{ud} \rangle, E_{ud} = \{\{u, d\} | (u, t, d) \in A\}$$

In words, an edge exists between a user and a document if the user has assigned the tag t to the document. The graph can once again be represented in matrix form, which we denote as $\mathbf{Y} = \{y_{ij}\}$, $y_{ij} = 1$ if there is an edge connecting u_i and d_j. This bipartite graph can be folded into two one-mode networks, which we denote as $\mathbf{S} = \mathbf{YY}'$, and $\mathbf{C} = \mathbf{Y}'\mathbf{Y}$.

The matrix $\mathbf{S}$ shows the affiliation between the users who have used the tag t, weighted by the number of documents to which they have both assigned the tag. Since a tag can be used to represent different concepts (such as *sf* for *San Francisco* or *Science Fiction*), and a document provides the necessary content to identify the contextual meaning of the tag, this network is likely to connect users who use the tag for the same meaning.

$\mathbf{C}$ can be considered as another angle of viewing the issue of polysemous or homonymous tags. Thus, with the edges weighted by the number of users who have assigned tag t to both documents, this network is likely to connect documents which are related to the same sense of the given tag.

3.3 Documents

Finally, a bipartite graph UT_d can also be obtained by considering a particular document d. The graph is defined as follows:

$$UT_d = \langle U \cup T, E_{ut} \rangle, E_{ut} = \{\{u, t\} | (u, t, d) \in A\}$$

In words, an edge exists between a user and a tag if the user has assigned the tag to the document d. The graph can be represented in matrix form, which we denote as $\mathbf{Z} = \{z_{ij}\}$, $z_{ij} = 1$ if there is an edge connecting u_i and t_j. Like in the cases of a single user and a single tag, this bipartite graph can be folded into two one-mode networks, which we denote as $\mathbf{M} = \mathbf{ZZ}'$, and $\mathbf{V} = \mathbf{Z}'\mathbf{Z}$.

The matrix $\mathbf{M}$ represent a network in which users are connected based on the documents commonly tagged by them. Since a document may provide more than one kind of information, and users do not interpret the content from a single perspective, the tags assigned by different users will be different, although tags related to the main theme of the document are likely to be used by most users. Hence, users linked to each other by edges of higher weights in this network are more likely to share a common perspective, or are more likely to concern a particular piece of information provided by the document.

On the other hand, the matrix $\mathbf{V}$ represents a network in which tags are connected and weighted by the number of users who have assigned them to the document. Hence, the network is likely to reveal the different perspective of the users from which they interpret the content of the document.

We can see that different relations between the users, the tags and the documents in a folksonomy will affect how a single user, tag or document is interpreted in the system. Each of these elements provide an appropriate context such that the semantics of the elements can be understood without ambiguity.

4 Semantics of Ambiguous Tags

One problem in the existing collaborative tagging system is the existence of ambiguous tags. By "ambiguous tags," we refer to tags that are intended to represent different concepts by the users. For example, in del.icio.us the tag *sf* has been used to describe documents which are related to science fiction and San Francisco. Another example is the tag *opera*, which are used for describing contents related to opera as a kind of musical performance as well as those related to the WWW browser which is named "Opera."[3]

As we have discussed, the semantics of a tag depends on the context given by the users who have used it as well as the documents being tagged. By studying the associations between the tag, the users and the documents, we may determine the different meanings of a tag by placing it in the right context. As an illustrative example, we present an analysis of the bipartite graphs obtained from a single tag, which we have chosen for its common occurrence and multiple equally-frequent meanings in order to preserve the clarity of the example. In particular, we would like to find out if it is possible to disambiguate a tag by studying its association with different users and documents.

4.1 Understanding a Single Tag

In the experiment described below, we try to examine the networks of users and documents associated with the tag *sf*, and attempt to understand how different interpretation of the tag can be discovered from the analysis of the networks.

The reasons of choosing the tag *sf* as an illustrating example are twofold. Firstly, *sf* is a tag used very frequently by users in del.icio.us. Although the exact number of times that the tag has been used cannot be known from the system, we are able to collect over 5000 triples which involves the tag *sf*. Secondly, by observation, the tag *sf* has been used by users to refer to two very distinctive concepts, namely "science fiction" and "San Francisco." We expect that users using the tag to refer to one of the two concepts do not use it to refer to the other one. Hence, the tag *sf* is more worthwhile to be examined, and we expect that experiments on the tag can produce clearer results for performing analysis.

In March 2007, data was collected from the del.icio.us website by using a crawler program written in Python. The program retrieved pages listing all bookmarks that have been tagged with *sf*, and subsequently retrieved the published RSS file of each bookmark to obtain the corresponding users and tags associated with it. In other words, the crawler retrieved bookmarks in del.icio.us which have

[3] http://www.opera.com/

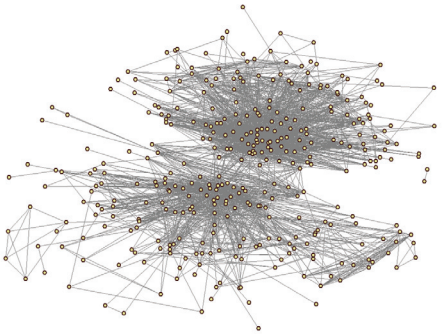


Fig. 1. A network of documents tagged by *sf*.

been tagged with *sf*, along with the users who tagged the page, and the tags, including *sf*, they used. In total, 238,117 triples were obtained, each involving a user, an URL of the bookmark, and a tag. A total of 427 distinctive URLs and 19979 users are involved. Out of these triples, 5852 involves the tag *sf*.

We extract all those triples that involve the tag *sf*, and construct the matrix $\mathbf{Y}$, representing the associations between users and bookmarks (documents). We then construct the matrices $\mathbf{S} = \mathbf{Y}\mathbf{Y}'$, corresponding to the network of users, and $\mathbf{C} = \mathbf{Y}'\mathbf{Y}$, corresponding to the network of documents.

The matrices $\mathbf{S}$ and $\mathbf{C}$ are feeded into the network analysis package Pajek [3], and visualized as networks. Since some users do not have any associations with other users, as in the case of documents, isolated nodes are removed from the networks. The results are shown in Fig 1 and Fig 2. In Fig 1, nodes represent documents, and two nodes are connected by an edge if a user has tagged both documents with the tag *sf*. Edges are weighted by the number of such users, and is not shown in the figure. In Fig 2, nodes represent users, and two nodes are connected by an edge if both users have tagged a document with the tag *sf*. Edges are weighted by the number of such documents. The networks are visualized using the Kamada-Kawai layout algorithm [8] implemented in Pajek.

Two large clusters of nodes can be observed in both of the networks in Fig 1 and Fig 2. However, as shown in the two figures, there are more connections between the two clusters in the network of documents than in that of users. One hypothesis that can be used to explain the existence of clusters in the network of documents is that they correspond to groups of documents related to the different senses of the tag *sf*. A similar hypothesis that can be applied to the network of users is that the different clusters corresponds to groups of users who have used the tag *sf* to represent different concepts.

Since documents are connected if a user tagged them with the tag *sf*, it implies that connected documents are considered by the user as all related to certain concept represented by the tag *sf*. In addition, if we assume that a user

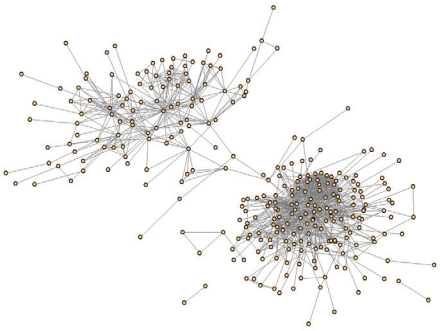


Fig. 2. The network of users who used the tag *sf*.

would be consistent in using the same tag for the same concept, it is reasonable to suggest that documents in different clusters would address a different concept represented by the tag *sf*. As we understand through observation that two major concepts – "science fiction" and "San Francisco" are associated with the tag *sf*, we can further suggest that the two major clusters in the network correspond to documents on science fiction and San Francisco respectively. To testify this hypothesis, we perform further analysis on the tagging data.

Firstly, we manually examine all the 357 websites represented by the nodes in the network of documents. We classify the websites into either related to science fiction or San Francisco, based on the content of the website as well as other tags used by the users. We indicate that the website cannot be classified into either of these categories if not enough information or evidence is available. After that, we combine the information with the original network, and use Pajek to draw a new network, as shown in Fig 3.

In the figure, circular nodes represent documents related to science fictions, and triangular nodes represent documents related to San Francisco. Documents that cannot be classified are represented by rectangular nodes. We can see that these two types of nodes are clearly grouped into two clusters. The result shows that the two clusters indeed correspond to two sets of documents related to two distinctive meaning of the tag *sf*.

However, it is interesting to note that there are actually a lot of edges connecting nodes from different clusters. Since nodes are connected if a user tagged them with the tag *sf*, these connections imply that some users actually used the same tag to represent two distinctive concepts. This also explains why the two clusters in the network of users are connected by a few edges. The documents connected by edges between clusters in the network of documents are then responsible for the edges connecting the users from different clusters in the network of users. However, since it would be very difficult to judge accurately whether

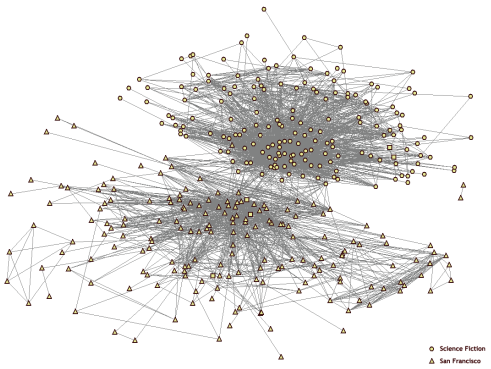


Fig. 3. The network of documents tagged by *sf* with classified nodes.

a user always uses the tag *sf* to refer to science fictions or San Francisco, we refrain from performing a similar classification of the users.

To further investigate whether there are many users who actually used the tag to refer to more than one concept, we construct one more network of documents. Based on the data which generates Fig 3, we remove edges which has a weight less than 2. By doing that we effectively ignore all the edges which correspond to cases in which only one user has used the tag *sf* on both of the documents connected by an edge. We also remove nodes that are not connected to any other nodes afterwards. The result is shown in Fig 4. it can be seen that there remains only one edge which connects nodes across the two clusters.

Finally, we examine how different tags are associated with each other given this set of documents and users. Since the documents are all tagged by the tag *sf*, all the other tags can be considered to be related to it. Given the two distinctive concepts represented by the tag, it is reasonable to hypothesize that the tags related to it can also be divided into two groups, one being related to science fictions, and another to San Francisco. We construct a matrix $\mathbf{T} = \{t_{ij}\}$ to represent the associations between the tags. t_{ij} is the number of times tag_i and tag_j have been used on the same document. Since there are over 8000 unique tags in the data, and many of them have been only used on a few documents, we only concentrate on 35 tags which are used most frequently along with *sf*. The associations between the tags are visualized in Fig 5. We can see that tags which are related to San Francisco are grouped in one cluster while tags related to science fictions are grouped in another cluster. This suggests that we can examine the related tags in order to obtain the different meanings of an ambiguous tag.

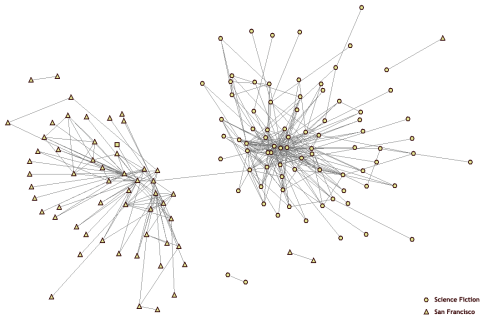


Fig. 4. The network of documents tagged by *sf* after removal of edges with weights less than two and unconnected nodes.

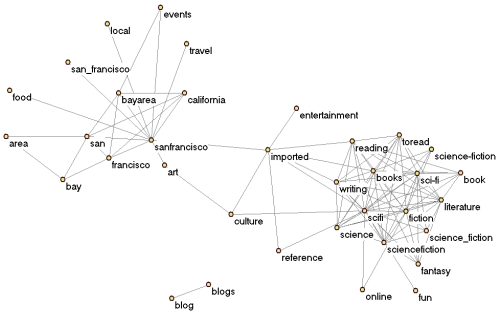


Fig. 5. The network of 35 tags which are most frequently used along with *sf*.

5 Discussions

The experiment results show that by analyzing the tripartite graph of folksonomy and the relations between tags, users and documents, we can discover how tags are being used, and better understand the meanings of the tags which are used for multiple meanings. Hence, although the same tag can be used to represent different concepts, the documents and the users still provide the context for understanding specific meanings of the tag. Given the above results, we come to understand more about the characteristics of folksonomies.

5.1 Ambiguous Tags from Users' Point of View

Based on the facts that documents of similar topics are clustered together, and that documents are connected by users who have applied the tag *sf*, we see that

the majority of users use the tag to refer to one concept only. This is because if users use the tag arbitrarily to refer to any of the two concepts, we would not be able to observe two clusters in the network. Hence, although a tag can possess several distinctive meanings, users tend to be consistent in referring to the same meaning when they use the tag. One may also suggest that users interested in one concept represented by the tag are not interested in the other, thus producing the two clusters of documents. However, given that the different senses of the tags we examined do not actually have conflicts with each other, and that the experiments actually involves quite a large number of users, it is more reasonable to suggest that consistence in usage is the reason of the clear distinction that we have observed. Hence, this shows that it is possible to understand whether a tag has multiple senses by examining the associations between users and documents.

5.2 Existence of Sub-communities

In the experiment, in addition to the two large clusters of nodes, we can also observe within the clusters that there are some nodes which tend to be grouped with each other to form smaller clusters. For example, in Fig 3 on the left and right ends of the clusters of triangular nodes, we can observe that some nodes are more connected with each other than with the rest of the nodes. This is probably because even if we consider all documents that are related to "San Francisco," there are still actually a wide range of documents related to different aspects of "San Francisco." If we look at the network of tags, we can see that tags related to "San Francisco" include *food*, *travel* and *culture*. Thus, these smaller clusters probably correspond to documents with more specific topics. More analysis will be performed in the future to verify this hypothesis.

5.3 Identifying the Topics of Documents

There are some documents (rectangular nodes in the network) which we cannot classify them into either the category of "science fiction" or "San Francisco." This is because either the documents are only very loosely related to one of these topics, or the tags associated with it are not indicative enough. However, as these rectangular nodes are located in one of the clusters we have observed, it becomes possible to judge, with high probability, the topics of these documents. Also, folksonomies reflect the classification scheme evolving from the collaborative effort of users. Hence, this judgement is not necessarily aligned with the intention of the author of the document. Rather, by saying that a document is related to a certain topic as judged by its location in the network, we are reflecting the opinions of the users. Thus, by constructing and examining the networks of documents, we are able to place the documents into the appropriate context, allowing us to understand what it is about from the viewpoint of users.

5.4 Related Works

Research on folksonomies mainly focuses on relations between tags instead of the semantics of individual tags. For example, Begelman et al. [2] propose an

automatic tag clustering algorithm to tackle the problem of synonyms. A more comprehensive method proposed by [14] is able to discover four different kinds of relations – relevant, conflicting, synonymous and unrelated – between tags. Mika [10] proposes to generate lightweight ontologies which are more meaningful by examining tag relations in the social context instead of studying their co-occurrences in documents. One piece of work which is closely related to topic presented here is that by Wu et al. [18], in which the authors investigate how emergent semantics can be derived from folksonomies. They employ statistical analysis on folksonomies, and study the conditional probabilities of tags in different conceptual dimensions. Tags with multiple meanings will then score high in more than one dimensions in the conceptual space. However, one limitation of their method is that the number of dimensions must be determined beforehand.

6 Conclusions and Future Work

Our study shows that mutual contextualization does occur among the three basic elements in a folksonomy, and that it is possible to acquire a better understanding of the semantics of ambiguous tags by constructing and studying the networks of documents and users associated with the tag.

Currently, many research works focus on how tagging data in folksonomies can be utilized to provide other services, such as identifying user interests, recommending relevant documents or constructing light-weight ontologies. However, all these applications require a better understanding of the semantics of tags in order to provide accurate and useful results. For example, it would not be wise to match users based on the tags they used without knowing that tags may possess different meanings. Hence, the work presented here can be considered as a first step to acquire a better understanding of folksonomies.

However, challenge remains in that while we can identify different groups of users and documents which correspond to different usage of an ambiguous tag, we still need other methods to integrate these different pieces of information to acquire the full picture. For example, how can we know, without examining every documents, which groups of users and documents are associated with a particular sense of a tag? This will be further investigated in our future work.

Specifically, in the future we will apply our method on other ambiguous tags to observe its performance. We hope to gain more insight on how to devise some automatic algorithms to perform tag meaning disambiguation. We will also study different methods of hierarchical clustering or community-discovering algorithms [4, 11], and investigate how these techniques can be applied to discover clusters of documents and users. It is hope that, by further examining the tags associated with different clusters, we can discover the different senses of a tag, probably by examining the tags being used most frequently in the clusters. Finally, we will extend our study to users as well as documents, and investigate how analysis on tripartite graphs can help discover useful information such as communities of users or clusters of documents with similar topics, which will be very useful in applications such as Web page recommendation or social network analysis.

References

1. Mathes Adam. Folksonomies - cooperative classification and communication through shared metadata. *http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html*, 2004.
2. Grigory Begelman, Philipp Keller, and Frank Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*, 2006.
3. Wouter de Nooy, Andrej Mrvar, and Vladimir Batagelj. *Exploratory Social Network Analysis with Pajek (Structural Analysis in the Social Sciences)*. Cambridge University Press, January 2005.
4. Michelle Girvan and M. E. J. Newman. Community structure in social and biological networks. *PROC.NATL.ACAD.SCI.USA*, 99:7821, 2002.
5. Thomas Gruber. Ontology of folksonomy: A mash-up of apples and oranges. http://tomgruber.org/writing/mtsr05-ontology-of-folksonomy.htm, 2005.
6. T. Hammond, T. Hannay, B. Lund, and J. Scott. Social bookmarking tools (i): A general review. *D-Lib Magazine*, 11(4), April 2005.
7. Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 411–426. Springer, June 2006.
8. T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.*, 31(1):7–15, 1989.
9. Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40, New York, NY, USA, 2006.
10. Peter Mika. Ontologies are us: A unified model of social networks and semantics. In *International Semantic Web Conference*, pages 522–536, 2005.
11. M.E.J. Newman. Analysis of weighted networks. *Physical Review E*, 70:056131, 2004.
12. Richard Newman. Tag ontology design. *http://www.holygoat.co.uk/projects/tags/*, 2004.
13. S. Niwa, Takuo Doi, and S. Honiden. Web page recommender system based on folksonomy mining for itng'06 submissions. In *ITNG 2006. Third International Conference on Information Technology: New Generations*, pages 388–393, 2006.
14. Satoshi Niwa, Takuo Doi, and Shinichi Honiden. Folksonomy tag organization method based on the tripartite graph analysis. In *IJCAI Workshop on Semantic Web for Collaborative Knowledge Acquisition*, January 2007.
15. Emanuele Quintarelli. Folksonomies: power to the people. ISKO Italy-UniMIB meeting, June 2005.
16. G. Smith. Atomiq: Folksonomy: Social classification. *http://atomiq.org/archives/2004/08/folksonomy_social_classification.html*, 2004.
17. Harris Wu, Mohammad Zubair, and Kurt Maly. Harvesting social knowledge from folksonomies. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 111–114, New York, NY, USA, 2006. ACM Press.
18. Xian Wu, Lei Zhang, and Yong Yu. Exploring social annotations for the semantic web. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 417–426, New York, NY, USA, 2006. ACM Press.

The 6th International Semantic Web Conference and
the 2nd Asian Semantic Web Conference

November 11~15 2007
BEXCO, Busan KOREA