COLLABORATIVE ONTOLOGIES AND ITS VISUALISATION IN CSCW SYSTEMS

Michael Vonrueden
University of Paderborn, Faculty of Electrical Engineering, Mathematics and Computer Science
Fuerstenallee 11, 33102 Paderborn, Germany

Thorsten Hampel
University of Paderborn, Faculty of Electrical Engineering, Mathematics and Computer Science
Fuerstenallee 11, 33102 Paderborn, Germany

Keywords: Collaborative Ontology Creation, Knowledge Representation, Userinterface, Semantic Web.

Abstract: The goal of semantic structures and especially the semantic web is to simplify knowledge retrieval in computer based systems. The Visual Cooperative Ontology Environment - short visCOntE - aims to support the process of a collaborative ontology creation and the mapping of an individual's mental map into a computer based knowledge representation. Due to the collaborative and graphical approach many requirements have to be considered to establish such a project. Beneath a deep description of visCOntE and possible usage scenarios, the question what is needed for a successful collaborative ontology creation and which functions a system should make available will be determined in detail.

1 INTRODUCTION

The problem of information overload is present in nearly every digital based system. The difficulty to gain an overview and to separate the wanted from the unwanted information, or more precisely, to transform it into personal knowledge is often an enormous cognitive performance. The latest ambitions of the W3C towards the semantic web shows a new way to handle these tasks (W3C, 2004a). The use of ontologies can make it easier to find, process and associate informations. But as an ontology is similar to a mental model of an individual, a monolithic mental model will show only a unique view to the world, which will, in some cases, be not comprehensibly for others. Therefore the creation of ontologies should cover the involvement of different parties to participate and construct an ontology in a collaborate process.

This paper outlines the problems with ontologies which are developed in a one to many relation and tries to find new ways for a collaborative creation of ontologies. After an analysis of what is needed for a collaborative ontology creation, the current development of a tool called visCOntE will be described. visCOntE (Visual Cooperative Ontology Environment) - part of the ^{open}sTeam (Structuring Information in Teams) open source environment, developed at the University of Paderborn - is designed to edit and browse ontologies in a collaborate environment.

The underlying high flexible ^{open}sTeam server architecture, provides manifold approaches to create and structure knowledge in a cooperative manner. Users can share and annotate their knowledge (e.g. Documents) or separate resp. order it by creating special rooms and access rights. ^{open}sTeam's communication facilities like the shared whiteboard supports the synchronous building of a cooperative information space (Hampel and Keil-Slawik, 2002). The visCOntE project is the very first step to integrate a new approach of ontology based knowledge management into this environment.

2 SEMANTIC STRUCTURES

A human sees the world not as a line up of single objects but rather as an interconnected ensemble. The ability to structure natural objects and artifacts in hierarchies and classes is one part of the preceding step in a construction of a semantic map. The other highly cognitive performance is to relate the items of a taxonomy to other items and to reason their associations. To achieve this relations a kind of logical reasoning has to be performed. With the resulting semantic structure new semantical integrations can be adopted. But the question is how to integrate such complex structures into a computer based system. The structure of an ontology allows the creation of special knowledge domains, which split the complex type of a human mental model into specialized parts, which

Ronaghi F. (2005).
INTEGRATED PERFORMANCE MANAGEMENT.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 294-299
DOI: 10.5220/0002551402940299

have a common base. These ontologies cover (instead of a whole representation) only a fine granulated part of it. This makes the processing of knowledge easier and provides more reusability for distributed ontologies.

3 COLLABORATION AND ONTOLOGIES

> ”The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.” (Tim Berners-Lee, 2001)

As mentioned above, ontologies are similar to an individuals mental model of a special domain and/or a view of the world. The process to create such a mental model is far more than a one-time procedure or a simple adaption of other individual's models. Moreover the development of a mental model is always in a state of continuous movement associated with social aspects like discussions, correction of wrong assessments or simply: questions and answers. The rule of a mental models creation could be described as followed: Starting from a generic point of knowledge the new adopted knowledge will become more and more special, but never reaches a final state.

This let us assume, that for a successful building of knowledge two parts are essential: continuousness and endlessness. The fact is, that this parts are not considered in ontologies which are created from one or a few authors for many users or recipients and those which claim a final state. Because of the non existing involvement of the future users and the missing social aspects, a collaborative approach is needed which will support the evolution of an ontology similar to a mental model.

A collaborative created ontology could bear advantages with regard to the acceptance and comprehension of a common knowledge domain by all parties. The ability to create, modify and contribute an individual knowledge could further affect the attendance of the whole process and the gain of new or renewed knowledge which come along with discussions. Moreover a hidden advantage exists in the interpretation of current states of an ontology, that could state out the actual knowledge of a community and outlines the way of creation up to an actual snapshot.

The resulting questions is, what should be considered to transform the idea of a collaborate creation of ontologies into a real word scenario. In relation to the previous conclusions, the following keywords should be reasonable represented in this scenario: visualization, comprehension, social exchange and usability.

4 visCOntE - VISUAL COOPERATIVE ONTOLOGY ENVIRONMENT

visCOntE - part of the ^{open}sTeam (Structuring Information in Teams) environment, developed at the University of Paderborn (Hampel and Keil-Slawik, 2002) - aims to be a web-based graphical front-end for the collaborative creation and modification of ontologies. Intended are three possible kinds of editable ontologies. Due to the capabilities of the ^{open}sTeam Server it is possible to provide a very global ontology for the whole (server-wide) community, the creation of special rooms and groups permit a shared ontology for related users and the personal user space for private one. In general visCOntE should act as the default application for files in the common W3C Ontologie-Format OWL (W3C, 2004b) , which makes it very flexible in importing existing ontologies. The environment itself is splitted into a client and a server module. The client, implemented in Scalable Vector Graphics, combines an ontology browser with directly accessible edit-components. With this browser it is possible to structure the taxonomy of the ontology and to add, remove and edit elements such as classes and individuals or their interconnection. The server module, responsible for storing and retrieving the ontology data in OWL Format, communicates to the client over a simplified XML-Format to avoid time consuming processes. The following outlines the in section *Collaboration and Ontologies* mentioned keywords in relation to the developed tool.

4.1 Server Module

Like the ^{open}sTeam server itself, the server-module is written entirely in the Pike Scripting Language. Since Version 7.6, the Pike API offers basic capabilities for the handling of RDF (Ora Lassila, 1999) and OWL structures (Michael K. Smith, 2004) . The visCOntE server-module builds up on this basic functionality by adding a more sophisticated and linked level which represents the structure of the low level OWL-Elements like Classes, Individuals and Properties. This linked structure extends the meta informations of these elements, so that a class for example knows which individuals are derived from itself or an individual can validate if it belongs to the inheritance line of a special class. A controllstructure acts as an intermediate between the elements and the textual representation on the one hand and the connection to the Pike API on the other. Further it handles the ontology access and versioning over platform specific storage handlers.

Due to the extended element representation the architecture supports the use of actions which can be performed on hierarchical and ontology specific data structures. These actions will take place only on the represented elements and not on the ontology itself, so moving a class to a new position in the tree, adding an individual of a class to the ontology or getting an individual with a specific property are possible without touching the existing data. The above mentioned ontology control object will save the changes only on a explicit save call. The execution of a single action produces a result object which is provided either as a simplified hierarchical XML Format or in a raw format of the above mentioned OWL-Elements. The action model can be extended by customized actions which makes it easier for custom interfaces to access and edit an ontology in a very special way and with various clients. With the package oriented design, the insertion of custom interface and the interoperable use of the resulting XML data structure the module is capable for other purposes or software projects.

4.2 Visualization

The Visualization of hierarchical structures like ontologies and the resulting large depth and/or width often consumes to much space. Concerning to Ben Shneiderman's Informations Seeking Mantra: *"Overview first, zoom and filter, then details on demand"* (Shneiderman, 1996) the browser's view is divided into two major parts to consider the dynamic structure of a collaborative created ontology. The treemap approach (Shneiderman, 1991) allows to get a fast and good overview of deep hierarchies. The

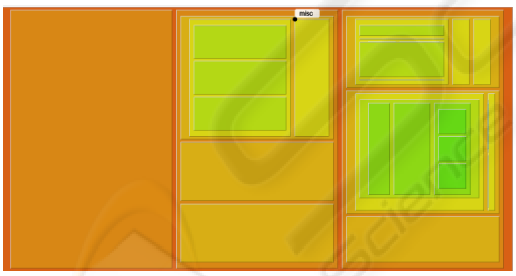

Figure 1: Treemap View

optimal space alignment and the quick navigation to deeper layers provide a good starting point to get a first overview. To distinguish the model's hierarchy, each tree-depth gets the same color, whereas the color differs only in a slight amount to it's parent color. The second visualization, the graph view, is related to a complex hierarchy. Especially ontologies with their associations, parent and child classes and attributes needs an easy respectively self-explanatory visualization. To understand the enormous complexity a deeper look on the construct of an ontology is required. An ontology according to the guidelines of

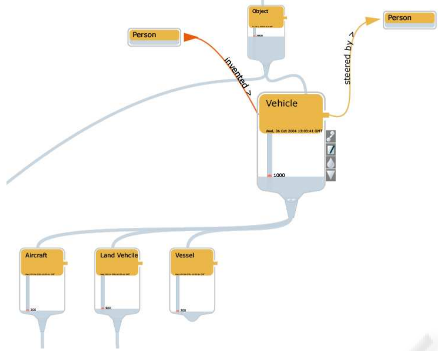

Figure 2: Graph of model with parent, childs and associations

the OWL-Format is structured into classes and individuals. A class represents a more generic description of an object, it links attributes and associations to itself and provides the inherited properties of its parent. On the other side an individual or instance represents a real world object, that is constructed from it's corresponding model. To achieve an easy to understand representation, we developed a metaphor which bears in mind the dependencies of the model-individual relation. The chosen metaphor maps the inheritance of the models as a fluid, which flows from one model to its submodel. The models layout looks like a kind of reservoir whose filling height indicates the number of individuals belonging to this model. If we proceed

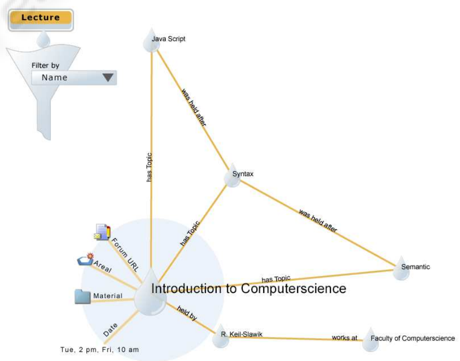

Figure 3: Individual View of Type "Lecture"

with this metaphor an individual could be seen as a part of the fluid, consequently this appears as a drop, which will show up in a zoomed view of the reservoir. Further the drops can be filtered by attributes and associations to achieve clusters with common properties. However the models attributes are shown as a box inside the reservoir, connections from this box to other models indicate the related associations. To keep the overview simple, only the current selected model, its direct parent and child models and the associated models are rendered.

Figure 4: Filtering of drops by different properties and associations

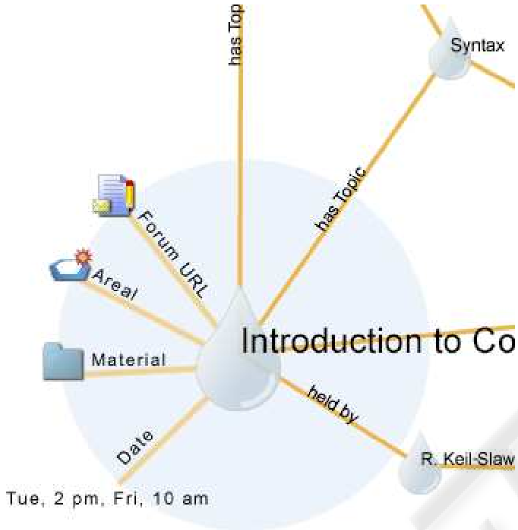

Figure 5: Individual in connection with properties and other individuals

4.3 Awareness

The understanding of how an ontology evolved to a current state is of particular importance. On the one hand side an involved author should see which changes are made to a model to recognize its current progress and what intention another editor's change belong to. On the other hand an important case is to provide a quick overview for future users, who participate the collaborative process at a later time. Therefore changes need to keep logged and facilities for asynchronous conversations are essential. Having much experience with Wiki-Systems we decided to adopt the approach of logging every changed version to provide the ability to return to a previous state. This technique also enables an author to fork a previous version for new editing. Another media function we will adopt is the asynchronous discussion related to a model. With this we hope to achieve an effect of continuous exchange between the users with the result of new respectively renewed views and in finding a consensus.

4.4 Usability

To achieve a good usability the graphical user interface is evaluated at specified milestones by students, which indeed will be the future users. The current framework of the in section *Visualization* mentioned views provides the following components.

Tabbed Views The treemap and the graph view are separated in different tabs. The selected model in either one of this views will also change the current model in the other tab. So a fast switching without a new search of the selected model accelerate the browsing of the ontology.

Bookmarks Adopted from World Wide Web - Browsers, the feature of setting a bookmark for a particular model provides a quick tie up to a previous session.

History A history navigation enables the user to keep track of the visited models in a current session.

Address Field Due to the hierarchical nature of an ontology an omnipresent virtual path (similar to a filesystem path) will be printed to get informations about the current location of a model. Further the models are selectable which could also be used for navigation purposes.

Zoomed Views Because of the lossless magnification of Scalable Vector Graphics, zoomable views are supported for the treemap and graph view and also a general magnification of the whole application which is especially useful for visually handicapped persons.

Directly Editable The editing is provided directly at the specified model. Symbols like a drop or an reservoir beneath the model indicate the resulting actions. The edit-forms itself are provided in HTML, due to the text input limitations of SVG. To create new associations a special copy stack which holds the selected models will be initialized over the whole session.

Related Filters As mentioned in section Visualization, a model related filter will be provided to cluster the individuals by specific attributes.

Automatic Linking Similar to the behavior of links in Wiki-Systems, special types of links will handled differently. Common WWW-links will be considered and a direct jump to the page will be performed. Links containing targets to pictures will be rendered as picture instead of the textual appearance. Special links to ^{open}sTeam items like users, documents, groups and rooms can be entered simply by its unique id instead of a complicated URL. Further it is planned to support the embedding of audible and visual documents.

4.5 Participation

What we want to achieve is acceptance and participation for the cooperative creation of an ontology. The challenge is to communicate the future user the advantage and gain for his own purposes. In case of ontologies stimuli could be to point out that the users individual knowledge will stay in conjunction to others knowledge or that personal artifacts like Documents can be embedded in a semantic structure to be useful for others and in the end for the user himself. It is surely not enough to just communicate it with words, than with a direct response of the whole system. Further attendance could be achieved by embedding the system in university lectures or other workflow's in organizations. Especially organizations could profit from sharing the knowledge of its members to keep it beyond the individuals quitting. But covering this in detail would go to far for this paper.

5 USAGE SCENARIOS

In this section an imaginable usage scenario could take place. The ^{open}sTeam is available for all faculties of the University of Paderborn. To mention a few, the range of faculties goes from culture studies to computer science over economics up to natural science. Faculties outside the computer science uses the system often for accompanying lectures of a semester. After a semester the rooms and groups more or less gets orphaned. The chances to create an ontology at an ongoing lecture are obvious. Either for following groups of the same faculty or for the chance of a real interdisciplinary study, a group related or global ontology could help to keep the knowledge persistent in the system, which in turn could help students to revise on present semantic linked documents or relations. For clarity, the following fictive example is given.

5.1 Connecting knowledge domains

Imagine three courses, which will be held with support of the ^{open}sTeam environment. The first courses topic is about how to write screenplays for documentations, the second deals with current development of robotics at the university and the third is about guidelines of university marketing. Now the three courses create independently from each other models for their topics in the global ontology. They embed documents related to a specific field, relate these fields and documents to the involved students and to other previous existing models of the ontology.

In the next semester an economic course should create a short movie about the current research of advanced technology at the university, which should be presented to current and future investors. Now the three above mentioned independent courses could help to access related informations in a quick way. Browsing the ontology to get informations about leadings technologies and how to present them in attractive and for investors adequate form will help to keep the focus on the real problems. Due to the semantic structure of the ontology the students can reason what documents are related for their tasks, who could be a possible contact person and what else is associated. With the experience related to the creation of the movie new associations could be developed, which possibly connects the field of robotic with the field of university marketing with the one of how to write a screenplay.

This example should be more or less assignable to any kind of organization which has either a broad or a narrow spectrum of fields. Because of the ability of every involved person to embed its experience and knowledge, the approach in resolving a problem with an already solved problem and the direct access of related informations are the big chance of an ontology. Facts like the constructive fashion and the never reached final state of an ontology makes the essential difference of a collaborative created ontology in contrast to an expert only one.

5.2 Mapping ^{open}sTeam structure

Another possible scenario is the automatic mapping of the ^{open}sTeam structure to an ontology. The interconnection of users, room and documents could be visualized in a more detailed and clearly arranged manner. The addition of a special type of model could initiate a pre defined action, that performs changes in both, the ontology and the system structure. A particular example in this case could be the administration of the users room privileges or the relation of documents to specific rooms.

6 RELATED WORKS

One of the inspirations for visCOntE were on the one hand Wiki Systems like its well known representative wikipedia[1]. Structures like the collaborative editing of linkable content as a main approach in visCOntE and other media elements like an asynchronous item related discussion or the different handling of URIs (like weblinks, images or email-addresses) are adopted. Moreover the versioning of the Ontology is much the same like it is in a Wiki - System, visCOntE even extends this technique with the ability

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

to work with a forked version of the ontology. On the other hand ontology editors like Protégé [2] and OntoEdit©[3] with their fine granulated form based approach of ontology creation were taken into account for the development. With the visual approach of ontology creation and the design goal to be a tool for non experts visCOntE differs from these editors.

7 CONCLUSION AND FURTHER WORK

Because of the independent format of the ontology, the adaption is not limited to the context of the browse and edit environment. Where visCOntE is the first step in creating, searching and editing of the ontology, other services could also operate on the provided informations of it. This could be either in the way of an automatic editing of the ontology by special reasonings or in an automatic support of user interests.

General services can operate on any kind of ontology regardless to the real information. An imaginable service or agent could observe a special model of the ontology which a user places to a list of subscribed interest. On any change he/she will be informed about the current addition. Related to the usage scenario this could be a new document dealing about robotics or a newly associated field. This could help, to keep the user's knowledge updated. A different service but with the same approach could be the detection of people, who have similar interests or problems. Especially in organization like universities this could help out to find new learning partners. Another approach could be the initiation of a keyword based semantical search engine that in future could be embedded in visCOntE. The mentioned scope is directed to the informations an ontology provides. Other services like a reminder of appointments can operate only if the ontology has a relation to time. To initiate these kind of services a special architecture which analyses the attributes of an ontology is needed. The OWL-Standard supports the assignment of special data types like dates but others are specified by the author of the ontology. The special structure of the OWL-Format in separating datatypes from the actual models makes it possible to provide such architecture, that dynamically watches attributes and relates them to special events. Perhaps, even a creation of services on the user side is possible to construct a set of rules with corresponding actions. The list could be widened by more particular examples. What we wanted to mention is the wide field of application of an ontology thats reasonably needs at first to be constructed and in particular in a cooperative way. Without this very first step, services like the above remain meaningless, because the user is constrained to embed himself in the whole process. The visContE project is the very first step to integrate a new approach of the semantic web into the ^{open}sTeam environment. Further steps could be, like mentioned above, the integration of semantic webservices, the sharing of the constructed ontology with other parties or the representation of system characteristics or other hierarchical based informations. Beside this, we are looking forward to get an overview on social aspects and patterns of the systems user composition to answer questions like which research fields are more distinctive than others or how the cooperation of interdisciplinary groups is possible / impossible. The development of visContE is still in an early phase of userinterface creation and implementation of the server-module.[4]

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

[3]http://www.ontoedit.de

[4]Visit http://steam.uni-paderborn.de/visconte/

REFERENCES

Hampel, T. and Keil-Slawik, R. (2002). **sTeam**: Structuring information in a team - distributed knowledge management in cooperative learning environments. *ACM Journal of Educational Resources in Computing Vol. 1, No. 2 (2002), 1-27*.

Michael K. Smith, Chris Welty, D. L. M. (2004). Owl web ontology language guide. `http://www.w3.org/TR/2004/REC-owl-guide-20040210/`.

Ora Lassila, R. R. S. (1999). Resource description framework (rdf) model and syntax specification. `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222`.

Shneiderman, B. (1991). Tree visualization with tree-maps: a 2-d space-filling approach. *ACM Transaction on Graphics , Vol. 11, No. 1 (1991), 92-99.*

Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualisation. In *Proceedings of the 1996 IEEE Symposium on Visual Language (VL'96), 336-343*.

Tim Berners-Lee, James Hendler, O. L. (2001). The semantic web. Scientific American `http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2`.

W3C (2004a). Semantic web. `http://www.w3c.org/2001/sw/`.

W3C (2004b). Web ontology language (owl). `http://www.w3.org/2004/OWL/`.