Building Ontologies for Knowledge Management Applications in Group Sessions

Doris Meier
Fraunhofer Institute for Industrial Engineering,
Nobelstrasse 12, 70569 Stuttgart, Germany
doris.meier@iao.fhg.de

Carsten Tautz
empolis Knowledge Management GmbH
Sauerwiesen 2, 67661 Kaiserslautern, Germany
Carsten.Tautz@empolis.com

Ralph Traphöner
empolis Knowledge Management GmbH
Sauerwiesen 2, 67661 Kaiserslautern, Germany
Ralph.Traphoener@empolis.com

Michael Wissen
Fraunhofer Institute for Industrial Engineering,
Nobelstrasse 12, 70569 Stuttgart, Germany
Michael.Wissen@iao.fhg.de

Jürgen Ziegler
Fraunhofer Institute for Industrial Engineering,
Nobelstrasse 12, 70569 Stuttgart, Germany
Jürgen.Ziegler @iao.fhg.de

ABSTRACT

This paper describes a system supporting the engineering of ontologies in group sessions. Here we use the term *ontology* as a synonym for the corporate vocabulary which is used as the basis for a knowledge management system. We discuss the MetaChart method for distributed groups working in creative sessions with intuitive system support. The groups may create information structures in these sessions and save their knowledge. We show the application of the method and finally outline the organization of the information.

KEYWORDS

Distributed Knowledge Capture, Group Knowledge Capture, Ontologies

INTRODUCTION

In today's economy, people get more and more specialized as the processes to be performed become increasingly knowledge-intensive. Consequently, knowledge is not "duplicated" among the company staff without a dedicated effort. Instead, specialized knowledge is "owned" by experts. To avoid needless double work (because some colleague has already solved the problem) and to avoid the unplanned "forgetting" of knowledge (e.g., if an expert leaves the company), knowledge needs to be shared explicitly. The sharing of knowledge becomes vital to the competitiveness of a company.

However, to be able to share knowledge effectively, it is most helpful to establish a common vocabulary. Without such a common vocabulary, it is difficult to share knowledge because not everybody in the company interprets the information in the same way. Hence, the corporate vocabulary is the basis for any knowledge management system. As such it acts as meta knowledge enabling and guiding both the indexing and recording of new documents as well as the search for existing documents. We use an object oriented representation for this vocabulary consisting of classes (1 for each kind of document) and types attributes. The value ranges of the attribute types are either numeric (integer, real, date, etc.) or enumerations. Enumerations are specified using concepts and keys where keys are the varois word forms in which a particular concept can appear in a document (e.g. keys "house", "houses", "maison", "Haus", "Häuser" for the concept "house"). This allows for automatic knowledge extraction from existing documents.

If it is specified explicitly and is used by a knowledge management system, it is also often referred to as an *ontology* (Gruber, 1995). Clearly, all stakeholders must be involved in defining such an ontology as it decides on the usefulness of the resulting knowledge management system (Tautz, 2001). Each stakeholder (sponsor, implementers, future users, etc.) has his own distinct objectives, interests, and views. Therefore, bringing together the ideas of all stakeholders and detailing them to the degree needed is often a formidable and time-consuming task. This is especially true for knowledge management systems because existing knowledge (often from many knowledge sources) needs to be identified and structured.

Yet, existing ontology engineering methods (for an overview see (López, 1999)) and authoring environments of existing knowledge management systems fail to support this process in its entirety. Although there are systems guiding the knowledge acquisition from users (e.g., Blythe et al., 2001) or supporting the collaborative construction of ontologies (e.g., Farguhar et al., 1996), group discussions are not supported in which an ontology is incrementally developed from vague ideas to a concrete and detailed structure.

In a typical group discussion, some areas will have very detailed descriptions in the beginning (e.g., if a database exists, its schema can be readily used as part of the ontology), while others will be vague (this is usually the case for knowledge that has not been captured explicitly so far). These pieces of information are then gradually detailed and extended until a complete ontology results.

The difficulties in engineering ontologies lies mainly in the transformation of human knowledge that is distributed among several stakeholders to structured knowledge that can be processed by computers. The complexity of this integration process requires a communication and cooperation model, which goes considerably beyond simple group support and takes into account the further use of elaborated results.

A successfull approach should support the followings tasks:

- Centralized as well as distributed group work
- Personalized view of the information
- Simultaneous work at the same objects and notification events

With this background in mind, we concentrate on finding an intuitive method for capturing knowledge about ontologies, that means about objects and their associations. Information objects generated in the context of group cooperation (concerning computer supported cooperative work see: Greif, 1988) may be used to derive object structures as well as meta activities. At the end of such a session we have a complete overview of the structure of the information.

SCENARIO

For the following parts of the paper, a simple scenario will be used to show the advantages of the MetaChart method. In this scenario, the task of a group of stakeholders is to identify and structure all existing types of knowledge within the company "Smartco". Smartco produces home entertainment equipment. The task shall focus on the exploitation of these knowledge sources for a knowledge management system which is about to be installed at Smartco. Four stakeholders are involved:

- The sponsor Susan (who can make a decision on whether to exploit a given knowledge source or not)
- The database expert Greg (who has detailed knowledge on the trouble ticket database)
- The project manager Sharon (who has managed already several projects and has a good overview of what kind of knowledge could be helpful in future projects)
- The implementer John (who is responsible for setting up the knowledge management system)

While Sharon and John have met personally for this discussion, Susan and Greg can join the meeting only remotely due to their tight schedule.

The goal is to come up with an object-oriented model which describes the ontology and can be used as the basis for the later knowledge management system.

CHARACTERISTICS OF GROUP SESSIONS

What we aim at is an enhanced support for the user in the early phase of information structuring. Here it is of utmost importance which interaction possibilities are available to the user and what results are obtained. Group based interaction possibilities for cooperative work in creative environments with their special needs are to be supported.

To work productively with a computer system in a creative group session, this system must fulfill some requirements. First, the system has to be highly intuitive. The user may not be bothered by struggling with the system. It must not cost more time to edit results within the system as it would need to edit it on plain paper. Furthermore, there should not be a high effort in learning to operate with the system, but every group member should be able to use it instantly.

Second, there should exist the possibility for different group members to personalize their view of the objects. For example, it could be that the session was started in a group situation, basing on a powerwall. After the session, every member takes the results with him. To make use of these results, he should be able to personalize his view of the structure, for example to emphasize on certain parts without losing the rest of the information.

Another requirement of a creative group session is the possibility to begin a session at a very low level and add step by step semantics to the structures. First, for example, one could outline the main objects that should be modeled. Then, associations between the objects like inheritance, aggregation or interconnections could be added. Some objects perhaps should be extracted from the session and being modeled in an own group session, whose results flow back to the originating session. It is important that the system is open to any desired kind of semantics structure that the group desires to build. So there is no specification of structures which the group must meet, but the group is completely free and not handicapped by any predefined structures they must fulfill.

Creativity and cooperation support do not solely depend on the development of software-based tools. The physical environment where these tools are applied with their interaction possibilities is of equal importance. The implementation of the MetaChart method does not only provide for the exclusive use on customary PCs but also on large interactive interfaces as, e.g., the powerwall "Interactive Wall" at the Fraunhofer IAO (Wissen et al., 2001) with their various types of interaction.

Emphasizing on the aspect of distributed work, it is of utmost importance that every group member can participate in the session. Certainly this has to be combined with traditional methods of distributed group work like video conferencing. Using MetaChart, it is furthermore not only possible to discuss with other group members, but also to work together on the same structures and to see instantly any change made.

OVERVIEW OF THE METACHART METHOD

The conception of MetaChart presented in this contribution finds its application in the preparation and exchange of information and information structures within groups. We aim at the creation of a work environment in which group members can intuitively prepare information contents and link them into structures as simple as possible.

The specific phases of building up information structures as well as generating information and ideas are of particular importance. In each of these phases we have several scenarios reflecting different kinds of group work. These scenarios consider cooperation taking place at a single location as well as in a distributed environment. According to model-view-controlling (Krasner and Pope, 1988), each member of the group may simultaneously work at the same model and therefore interact from any place on a shared work space. Modifications of the model layer, e.g. adding attributes, creating links or structuring information objects, are visible to all participants. The distributed and therefore for all team members accessible work space simplifies the process of building up information as a basis for organizational and structural tasks.

Supplementing the approach of supporting group work is the possibility to us the MetaChart Method in a distributed environment. A group member which doesn't have the possibility to join in a session personally may connect himself via Internet to a running session and interact directly with the group, working actively and getting all changes instantly, as well as view the results later on for example in the included web interface.

The work surface does not only serve as a pure media of presentation but, in fact, it simplifies the process of information building as a central platform for organizational and structural activities in combination with different input and interaction possibilities, fulfilling the requirements of a group session as mentioned above.

USER INTERFACE

As outlined before, the user interface has to be highly flexible and intuitive to be accepted as a proper instrument during creative group sessions. For the intuitivity there are two important aspects:

1. the GUI must meet the modern requirements of usability engineering
2. the methods to input data must match the special requirements of group sessions

The MetaChart Application offers different suitable methods for gathering input like recognizing of mouse gestures, or character recognizing. Furthermore, while working in a session, arbitrary data can be added to that session. Objects, which stand in associations to other objects, may content as different formats as plain text, images, web links, results of program calls and so on.

To point out a typical work situation, we use the scenario to show how MetaChart can be used.

- Greg imports the schema of the trouble ticket database resulting in a list of attributes: organization and contact information who sent trouble ticket, date of trouble ticket, short title, description, date when problem was solved
- John argues that “date when problem was solved” does not make sense. Instead, it should be captured how long it takes until a trouble ticket has been closed.
- Susan adds that the solution should be added to the trouble ticket. Up to now, problem and solution were not stored together.
- John intervenes that trouble tickets need to be further characterized to enable meaningful search.
- Sharon remembers that she received several solution reports for her trouble tickets and opens three exemplary solution reports.
- John and Sharon analyze the solution reports for commonalities and differences to come up with additional attributes.
- They realize that (a) a trouble ticket may have several (alternative) solutions and (b) a solution may be remedy for several trouble tickets. This calls for a n:m relationship.
- Sharon suggests a taxonomy for characterizing trouble tickets.
- Meanwhile Greg queries the database to retrieve the most recent 20 trouble tickets.
- Sharon and Greg try to classify those according to Sharon’s taxonomy.
- In parallel, John analyzes the reports further and marks several words in the texts. These constitute new objects. Example for report (excerpt): “To ensure that the *screw* does not gradually become loose, do not reinsert the screw to fix the *cassette mechanics*. Rather *glue* it. …” (italic words are those marked by John – they should become part of the ontology)

In MetaChart, the following screenshot (Figure 1) could be a result of the group session.

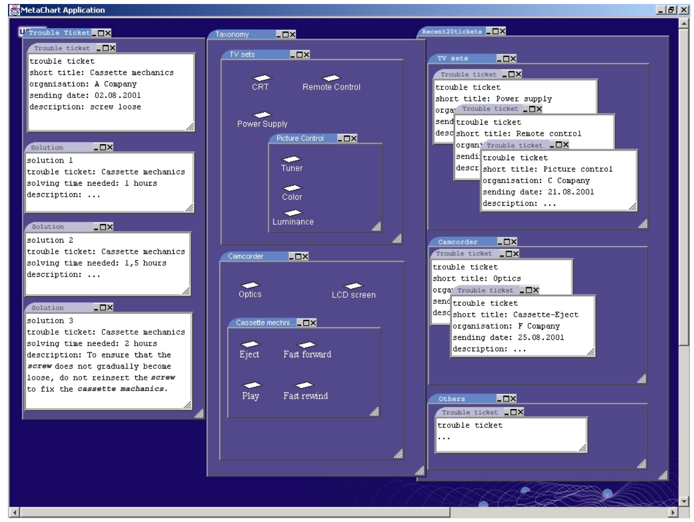

Figure 1: Snapshot of MetaChart Application

ORGANISATION OF INFORMATION

For implementing knowledge management solutions, we use orenge, a software system by empolis[1] that is specifically designed for the effective and efficient structuring and retrieval of already existing knowledge sources, e.g., a collection of documents or databases. orenge’s behavior is based on a meta-model. Each knowledge source is described by a set of attribute value pairs. The meta-model defines which set of attributes is to be used for each type of knowledge source. Such a definition is called an *aggregate*. Therefore, documents can be organized into classes where each class has its own characteristic set of attributes or aggregate.

All attributes of the meta-model are typed. In orenge an extensive array of predefined types is available which can be further detailed for a specific application. One of the most common specializations concerns the Text type. It is typically done by enumerating possible values called *concepts* for its range. The taxonomy used for the trouble tickets is an example. Of course, such a scheme would have to be agreed on by the stakeholders of a knowledge management system.

This basic meta-model is enhanced by several other models which describe various aspects of the final knowledge management application. At this point, only two shall be described in detail:

1. The *import model* describes which databases or documents should be analysed for structuring the already available information. In addition, it defines which columns (in case of databases) or

[1] www.empolis.com

which text sections (in case of documents) will be used for analysis. For example, the section labelled "Affected Product" could be used to analyse the product a trouble ticket is about.

2. The *analysis model* describes the linguistic analysis of knowledge sources. The analysis automatically extracts the values for the aggregate's attributes from a textual description. One way of identifying values is using so-called *keys*. For example, if the keys "TV", "TV set", "television", and "television set" were associated with the concept "TV", a trouble ticket would be associated to the product "TV" if one of these keys would appear in the "Affected Product" section.

To define a complete meta-model, empolis conducts modelling workshops with its customers. The experience in these workshops has been that attributes, types (incl. ranges), keys, and import models tend not to be defined in a sequential order but rather based on whatever comes to mind first. Therefore, a knowledge capturing approach as it is supported by the MetaChart approach is very helpful.

In contrast to the orenge model with its strictly defined semantics, the MetaChart data model is more generic. It allows defining objects and relationships between these objects. Objects (visualized by a rectangle) can be arranged graphically in two ways: One rectangle can be contained in another or two rectangles can be associated via an arrow. In addition, each rectangle can be associated with a set of attributes (content) and meta data (describing data). An export tool interprets the MetaChart and generates an initial (meaningful) orenge meta-model.

In the MetaChart application, there is not a lot of pre-built semantics in the data model. The semantics has to be added while transforming it to an ontology. For example, an hierarchical relation between two objects can mean inheritance or aggregation, at the moment of creation for the sytem it is of no importance. Later in the process of transformation it is specified exactly how the objects relate to each other. For our purpose, we defined the following translation rules among others:

- One meta-attribute of each object describes its type (aggregate, attribute, type, concept, import model, or knowledge source).
- The containment relationship is interpreted according to its elements. For example, if the surrounding object is typed as an aggregate, its elements are interpreted as attributes. If the surrounding object is typed as a type and its inner elements are typed as concepts, the inner elements constitute the range of the type. If the inner elements are also typed as types, the union of the ranges of the inner elements will be taken as the range of the surrounding type (corresponds to orenge's "compound type").

FURTHER PROCESSING OF THE CONTENT IN ORENGE

Once the initial meta-model has been created using the MetaChart tool, it is exported as an orenge model. This model can then be further enhanced using orenge's authoring environment tengerine. A knowledge engineer completes the model by various aspects, thereby defining the retrieval behavior of the resulting knowledge management system. Some of the most important aspects include:

1. the *similarity model* which describes how to compute the relevance of a given knowledge source to a user's query. This enables orenge not only to return exact matches, but also "almost matching, but still interesting" hits.
2. *explanations* describing why a particular knowledge source is part of the search result (which is not always obvious if a system not only returns 100% matches).
3. *dialog strategy* which defines how to guide the user to attain reasonable results as quickly as possible
4. *rules* which describe how to complete a user query (automatic inference or correction of query values based on the information given by the user)

CONCLUSIONS

Experience has shown that the construction of ontologies for knowledge management systems (KMS) must consider the input of all stakeholders of the later KMS. However, to our knowledge, group based knowledge capturing tools for this purpose supporting the process of going from vague ideas to complete structured models do not exist yet.

The outlined MetaChart approach supports this process allowing to structure information and content of cooperative work in the context of group sessions. Generated information objects and their incidental structures can be explored and modified in a user-specific and web-based way. Furthermore, this method offers an approach to coordinate tasks generated within a group session, and it offers also the possibility of collaborative system design.

Therefore, we have adopted this approach as a front-end authoring system for initial meta-models of orenge, a knowledge management system by empolis which

- automatically extracts meta-information from existing knowledge sources using a meta-model and

- retrieves knowledge based on the meta-model and the extracted meta-information.

Currently, we are developing the export module for the MetaChart tool to validate the improvement in tool support for the development of group based meta-models for knowledge management applications.

Besides the advantage to define a model in a group session, other major benefits of the MetaChart approach lie in its possibility to do late "typing" and structuring. For example, a list of words can be thought of at first without specifying whether this list is to be interpreted as a set of attributes (making up an aggregate describing a knowledge source), a set of concepts (making up the range of a text attribute), or a set of terms (making up the list of possible occurrences of a concept). Only later in the design process, the group needs to decide how to type the objects and how they relate to each other. Typing and structuring is done on an incremental basis as the stakeholders understand the domain increasingly better (based on their discussion of the MetaChart). Typing and structuring can be deferred until the MetaChart is exported as an initial orenge model. empolis found these two features to be vital during modeling workshops it conducts with its customers as experts tend to jump between subject areas and (at least at first) do not want to be bothered with technical details.

ACKNOWLEDGEMENTES

This work has been conducted in context of the German Lead Project INVITE with support of the German Ministry of Education and Research.

REFERENCES

[1] Blythe, J.; Kim, J.; Ramachandran, S.; Gil, Y., "An Integrated Environment for Knowledge Acquisition", In: Proceedings of the International Conference on Intelligent User Interfaces, Santa Fe, New Mexico, January 2001.

[2] Farquhar, A.; Fikes, R.; Rice, J., "The Ontolingua Server: A Tool for Collaborative Ontology Construction", In: Proceedings of the Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Alberta, 1996.

[3] Greif, I., "Computer-Supported Cooperative Work", A Book of Readings. San Marco (CA): Morgan Kaufmann, 1988.

[4] Gruber, T.R., "Toward principles for the design of ontologies used for knowledge sharing". International Journal on Human-Computer Studies, vol. 43, 1995, P. 907–928.

[5] Krasner, G.; Pope, S., "A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80.", In: JOOP 8/9, P. 26ff., 1988

[6] López, M.F., "Overview of the methodologies for building ontologies". In V. R. Benjamins, B. Chandrasekaran, A. Gómez-Pérez, N. Guarino, and M. Uschold (Eds): Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, August 1999.

[7] Sommestad, G., "The Literary Machine", URL http://www.sommestad.com/lm.htm.

[8] Tautz, C., *Customizing Software Engineering Experience Management Systems to Organizational Needs*, PhD Thesis, Department of Computer Science, University of Kaiserslautern, Fraunhofer IRB, Stuttgart, Germany, 2001.

[9] Wissen, M; Wischy, M. A.; Ziegler, J., "Realisierung einer laserbasierten Interaktionstechnik für Projektionswände.", In J.Krause (Ed.): Mensch & Computer 2001, P. 135-143, Teubner, Stuttgart, 2001.