Methodology for Development and Employment of Ontology based Knowledge Management Applications

York Sure
Institute AIFB
University of Karlsruhe
76128 Karlsruhe, Germany
http://www.aifb.uni-karlsruhe.de
sure@aifb.uni-karlsruhe.de

Steffen Staab*
Institute AIFB
University of Karlsruhe
76128 Karlsruhe, Germany
http://www.aifb.uni-karlsruhe.de
staab@aifb.uni-karlsruhe.de

Rudi Studer†
Institute AIFB
University of Karlsruhe
76128 Karlsruhe, Germany
http://www.aifb.uni-karlsruhe.de
studer@aifb.uni-karlsruhe.de

ABSTRACT

In this article we illustrate a methodology for introducing and maintaining ontology based knowledge management applications into enterprises with a focus on Knowledge Processes and Knowledge Meta Processes. While the former process circles around the usage of ontologies, the latter process guides their initial set up. We illustrate our methodology by an example from a case study on skills management.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Ontology, Knowledge Process, Knowledge Meta Process

1. INTRODUCTION

In recent years Knowledge Management (KM) has become an important success factor for enterprises. Increasing product complexity, globalization, virtual organizations or customer orientation are developments that ask for a thorough and systematic management of knowledge – within an enterprise and between several cooperating enterprises. Obviously, KM is a major issue for human resource management, enterprise organization and enterprise culture – nevertheless, information technology (IT) plays the crucial enabler for many aspects of KM. As a consequence, KM is an inherently interdisciplinary subject.

IT-supported KM solutions are built around some kind of organizational memory [1] that integrates informal, semiformal and formal knowledge in order to facilitate its access, sharing and reuse by members of the organization(s) for solving their individual or collective tasks [5]. In such a context, knowledge has to be modelled, appropriately structured and interlinked for supporting its flexible integration and its personalized presentation to the consumer. Ontologies have shown to be the right answer to these structuring and modeling problems by providing a formal conceptualization of a particular domain that is shared by a group of people in an organization [11, 6].

There exist various proposals for methodologies that support the systematic introduction of KM solutions into enterprises. One of the most prominent methodologies is CommonKADS that puts emphasis on an early feasibility study as well as on constructing several models that capture different kinds of knowledge needed for realizing a KM solution [13]. Typically, these methodologies conflate two processes that should be kept separate in order to achieve a clear identification of issues [14]: whereas the first process addresses aspects of introducing a new KM solution into an enterprise as well as maintaining it (the so-called "Knowledge Meta Process"), the second process addresses the handling of the already set-up KM solution (the so-called "Knowledge Process") (see Figure 1). *E.g.* in the approach described in [12], one can see the mixture of aspects from the different roles that, *e.g.* "knowledge identification" and "knowledge creation" play. The Knowledge Meta Process would certainly have its focus on knowledge identification and the Knowledge Process would rather stress knowledge creation. However, Knowledge Management is a process which is not only governed by IT. Hence, one needs to keep the balance between human problem solving and automated IT solutions. This balancing distinguishes KM from traditional knowledge-based systems.

*Additional affiliation:
Ontoprise GmbH, Karlsruhe, Germany, http://www.ontoprise.de

†Additional affiliations:
Ontoprise GmbH, Karlsruhe, Germany, http://www.ontoprise.de
FZI Research Center for Information Technologies, KM Department (WIM), Karlsruhe, Germany, http://www.fzi.de/wim
L3S Learning Lab, Hannover/Karlsruhe, Germany, http://www.learninglab.de

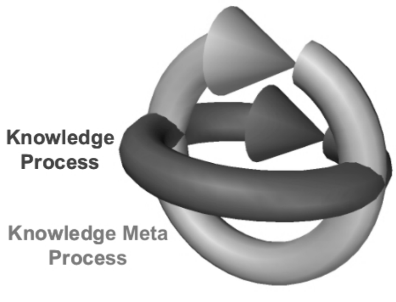

Figure 1: Two orthogonal Processes with Feedback Loops

The here presented methodology was developed and applied in the EU project On-To-Knowledge[1] [4]. In this paper we describe the knowledge meta process and the knowledge process and illustrate the instantiation of the knowledge meta process by an example from a skills management case study of the On-To-Knowledge project.

2. KNOWLEDGE META PROCESS

The Knowledge Meta Process (*cf.* Figure 2) consists of five main steps. Each step has numerous sub-steps, requires a main decision to be taken at the end and results in a specific outcome. The main stream indicates steps (phases) that finally lead to an ontology based KM application. The phases are “Feasibility Study”, “Kickoff”, “Refinement”, “Evaluation” and “Application & Evolution”. Below every box depicting a phase the most important sub-steps are listed, *e.g.* “Refinement” consists of the sub-steps “Extract knowledge” and “Formalize” etc.. Each document-flag above a phase indicates major outcomes of the step, *e.g.* “Kickoff” results in an “Ontology Requirements Specification Document” etc. Each node above a flag represents the major decisions that have to be taken at the end to proceed to the next phase. The major outcomes typically serve as decision support for the decisions to be taken. The phases “Refinement – Evaluation – Application & Evolution” typically need to be performed in iterative cycles. One might notice that the development of such an application is also driven by other processes, *e.g.* software engineering and human issues. We will only briefly mention some human issues in the example section.

2.1 Feasibility Study

Any knowledge management system may function properly only if it is seamlessly integrated in the organization in which it is operational. Many factors other than technology determine success or failure of such a system. To analyze these factors, we initially start with a *feasibility study* [13], *e.g.* to identify problem/opportunity areas and potential solutions. In general, a feasibility study serves as a decision support for economical, technical and project feasibility, determining the most promising focus area and target solution.

2.2 Kickoff

In the kickoff phase the actual development of the ontology begins. Similar to requirements engineering and as proposed by [10] we start with an **ontology requirements specification document** (ORSD). The ORSD describes what an ontology should support, sketching the planned area of the ontology application and listing, *e.g.* valuable knowledge sources for the gathering of the semi-formal description of the ontology. The ORSD should guide an ontology engineer to decide about inclusion and exclusion of concepts and relations and the hierarchical structure of the ontology. In this early stage one should look for already developed and potentially reusable ontologies.

The **outcome** of this phase is (beside the ontology requirement specification document (ORSD)) a semi-formal description of the ontology, *i.e.* a graph of named nodes and (un-)named, (un-)directed edges, both of which may be linked with further descriptive text *e.g.* in form of mind maps (*cf.* [2, 16]). If the requirements are sufficiently captured, one may proceed with the next phase. The **decision** is typically taken by ontology engineers in collaboration with domain experts. “Sufficiently” in this context means, that from the current perspective there is no need to proceed with capturing or analyzing knowledge. However, it might be the case that in later stages gaps are recognized. Therefore, the ontology development process is cyclic.

2.3 Refinement

During the kick-off and refinement phase one might distinguish in general two concurrent approaches for modeling, in particular for **knowledge extraction** from relevant knowledge sources: top-down and bottom-up. In a **top-down**-approach for modeling the domain one starts by modeling concepts and relationships on a very generic level. Subsequently these items are refined. This approach is typically done manually and leads to a high-quality engineered ontology. Available top-level ontologies may here be reused and serve as a starting point to develop new ontologies. In our example scenrio we encountered a **middle-out** approach, *i.e.* to identify the most important concepts which will then be used to obtain the remainder of the hierarchy by generalization and specialization. However, with the support of an automatic document analysis, a typical **bottom-up**-approach may be applied. There, relevant concepts are extracted semi-automatically from available documents. Based on the assumption that most concepts and conceptual structures of the domain as well the company terminology are described in documents, applying knowledge acquisition from text for ontology design helps building ontologies automatically.

To **formalize** the initial semi-formal description of the ontology ontology engineers firstly form a taxonomy out of the semi-formal description of the ontology and add relations other than the “is-a” relation which forms the taxonomical structure. The ontology engineer adds different types of relations as analyzed *e.g.* in the competency questions to the taxonomic hierarchy. However, this step is cyclic in itself, meaning that the ontology engineer now may start to interview domain experts again and use the already formalized ontology as a base for discussions. It might be helpful to visualize the taxonomic hierarchy and give the domain experts the task to add attributes to concepts and to draw relations between concepts (*e.g.* we presented them the taxonomy in form of a mind map as mentioned in the previous section). The ontology engineer should extensively document the ad-

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

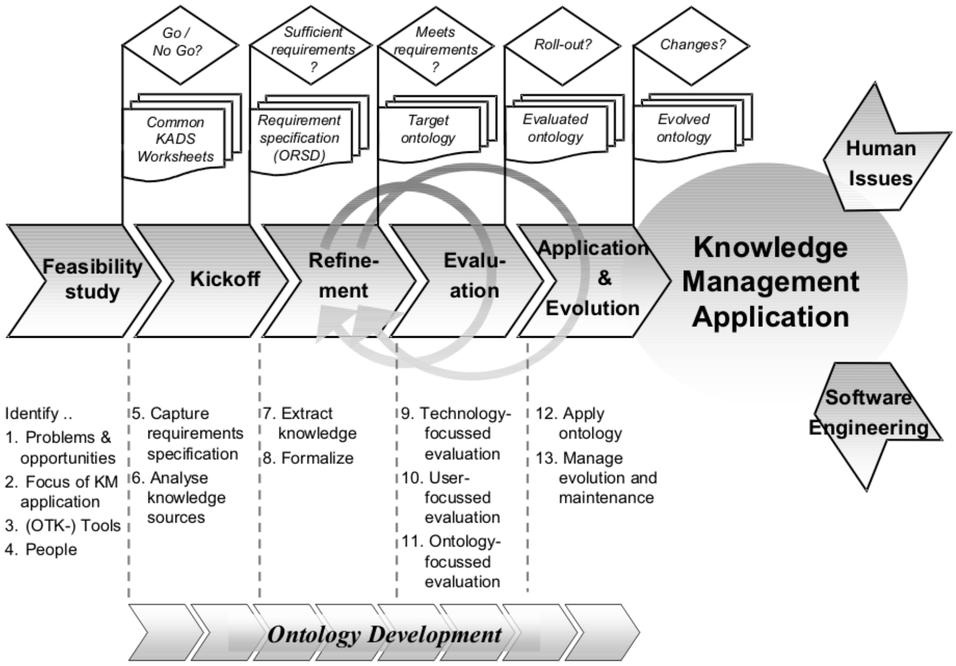

Figure 2: The Knowledge Meta Process

ditions and remarks to make ontological commitments made during the design explicit.

The **outcome** of this phase is the "target ontology", that needs to be evaluated in the next step. The major **decision** that needs to be taken to finalize this step is whether the target ontology fulfills the requirements captured in the previous kickoff phase. Typically an ontology engineer compares the initial requirements with the current status of the ontology. This decision will typically be based on the personal experience of ontology engineers. As a good rule of thumb we discovered that the first ontology should provide enough "flesh" to build a prototypical application. This application should be able to serve as a first prototype system for evaluation.

2.4 Evaluation

We distinguish between three different types of evaluation: (i) technology-focussed evaluation, (ii) user-focussed evaluation and (iii) ontology-focused evaluation.

Our evaluation framework for **technology-focussed evaluation** consists of two main aspects: (i) the evaluation of properties of ontologies generated by development tools, (ii) the evaluation of the technology properties, i.e. tools and applications which includes the evaluation of the evaluation tool properties themselves. In an overview these aspects are structured as follows:(i) Ontology properties (*e.g.* language conformity (Syntax), consistency (Semantics)) and (ii) technology properties (*e.g.* interoperability, turn around ability, scalability etc.).

The framework shown above concentrates on the technical aspects of ontologies and related ontologies. However, the aspect of **user-focussed evaluation** remains open. The most important point from our perspective is to evaluate whether users are satisfied by the KM application. More specific, whether an ontology based application is at least as good as already existing applications that solve similar tasks.

Beside the above mentioned process oriented and pragmatic evaluation methods, one also need to **formally evaluate ontologies**. One of the most prominent approaches here is the OntoClean approach (*cf.* *e.g.* [7]), which is based on philosophical notions. Applying this approach leads to more correct hierarchies of ontologies.

The **outcome** of this phase is an evaluated ontology, ready for the roll-out into a productive system. However, based on our own experiences we expect in most cases several iterations of "Evaluation – Refinement – Evaluation" until the outcome supports the decision to roll-out the application. The major **decision** that needs to be taken for finalizing this phase is whether the evaluated ontology fulfills all evaluation criteria relevant for the envisaged application of the ontology.

2.5 Application & Evolution

The **application** of ontologies in productive systems, or, more specifically, the usage of ontology based systems, is being described in the following Section 3 that illustrates the knowledge process.

The **evolution** of ontologies is primarily an organizational process. There have to be strict rules to the update, insert and delete processes of ontologies (*cf.* [15]). We recommend, that ontology engineers gather changes to the ontology and initiate the switch-over to a new version of the ontology after thoroughly testing all possible effects to the application. Most important is therefore to clarify *who* is responsible for maintenance and *how* it is performed and in *which time in-*

tervals is the ontology maintained.

The **outcome** of an evolution cycle is an evolved ontology, *i.e.* typically another version of it. The major **decision** to be taken is when to initiate another evolution cycle for the ontology.

3. KNOWLEDGE PROCESS

Once a KM application is fully implemented in an organization, knowledge processes essentially circle around the following steps (*cf.* Figure 3).

- *Knowledge creation* and/or *import* of documents and meta data, *i.e.* contents need to be created or converted such that they fit the conventions of the company, *e.g.* to the knowledge management infrastructure of the organization;
- then knowledge items have to be *captured* in order to elucidate importance or interlinkage, *e.g.* the linkage to conventionalized vocabulary of the company by the creation of relational metadata;
- *retrieval of* and *access to knowledge* satisfies the "simple" requests for knowledge by the knowledge worker;
- typically, however, the knowledge worker will not only recall knowledge items, but she will process it for further *use* in her context.

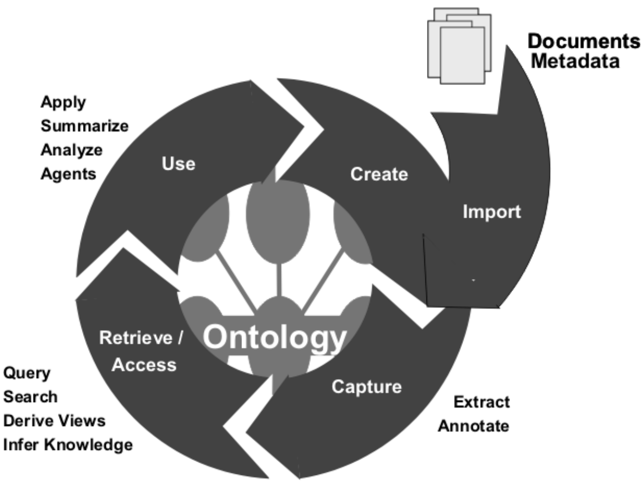

Figure 3: The Knowledge Process

4. EXAMPLE: SKILLS MANAGEMENT @ SWISS LIFE

We now give an example of the Knowledge Meta Process instantiation of a skills management case study at Swiss Life (*cf.* [9]). Skills management makes skills of employees explicit. Within the case study existing skill databases and documents (like *e.g.* personal homepages) are integrated and expanded. Two aspects are covered by the case study: first, explicit skills allow for an advanced expert search within the intranet. Second, one might explore his/her future career path by matching current skill profiles *vs.* job profiles. To ensure that all integrated knowledge sources are used in the same way, ontologies are used as a common mean of interchange to face two major challenges. Firstly, being an international company located in Switzerland, Swiss Life has internally four official languages, *viz.* German, English, French and Italian. Secondly, there exist several spellings of same concepts, *e.g.* "WinWord" *vs.* "MS Word". To tackle these problems, ontologies offer external representations for different languages and allow for representation of synonymity. Figure 4 shows a screenshot from the skills management application. The prototype enables any employee to integrate personal data from numerous distributed and heterogeneous sources into a single coherent personal homepage.

4.1 Feasibility Study

For identifying factors which can be central for the success or failure of the ontology development and usage we made a requirement analysis of the existing skills management environment and evaluated the needs for a new skills management system. We identified mainly the human resources department and the management level of all other departments as actors and stakeholders for the skills management. After finding the actors and stakeholders in the skills management area, we named the ontology experts for each department, which are preferably from the associated training group of each department.

4.2 Kickoff

The departments private insurance, human resources and IT constitute three different domains that were the starting point for an initial prototype. The task was to develop a skills ontology for the departments containing three trees, *viz.* for each department one. The three trees should be combined under one root with cross-links in between. The root node is the abstract concept "skills" (which means in German "Kenntnisse/Faehigkeiten") and is the starting point to navigate through the skills tree from the top.

During the **kickoff** phase two workshops with three domain experts[2] were held. The first one introduced the domain experts to the ideas of ontologies. Additional potential knowledge sources were identified by the domain experts, that were exhaustively used for the development of the ontologies, *e.g.* a book of the Swiss Association of Data Processing ("Schweizerischer Verband fuer Datenverarbeitung") describing professions in the computing area in a systematic way similar to an ontology. Obviously, this was an excellent basis to manually build the skills ontology for the IT domain. First experiments with extracting an ontology semi-automatically by using information extraction tools did not satisfy the needs for a clearly structured and easily understandable model of the skills. The domain experts and potential users felt very uncomfortable with the extracted structures and rather chose to build the ontology by themselves "manually". To develop the first versions of the ontologies, we used a mind mapping tool ("MindManager"). It is typically used for brainstorming sessions and provides simple facilities for modelling hierarchies very quickly. The early modelling stages for ontologies contain elements from such brainstorming sessions (*e.g.* the gathering of the semiformal ontology description).

[2]Thanks to Urs Gisler, Valentin Schoeb and Patrick Shann from Swiss Life for their efforts during the ontology mod-

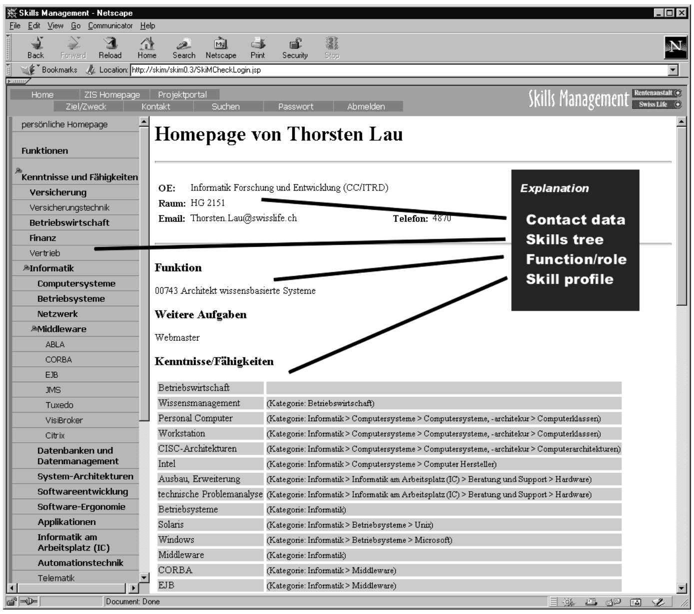

Figure 4: Skills Management Case Study @ Swiss Life

During this stage a lot of "concept islands" were developed, which were isolated sets of related terms. These islands are subdomains of the corresponding domain and are self-contained parts like "operating systems" as sub domain in the IT domain. After developing these concept islands it was necessary to combine them into a single tree. This was a more difficult part than assembling the islands, because the islands were interlaced and for some islands it was possible to add them to more than one other island, which implies awkward skills trees that contain inconsistencies after merging. For each department one skills tree was built in separate workshops. A problem that came up very early was the question where to draw the line between concepts and instances. *E.g.* is the programming language Java instantiated by "jdk1.3" or is "jdk1.3" so generic that it still belongs to the concept-hierarchy? Another problem was the size of the ontology. What is the best depth and width of each skills tree? Our solution was, that it depends on the domain and should be determined by the domain expert.

As **result** of the kick-off phase we obtained the semi-formal ontology descriptions for the three skills trees, which were ready to be formalized and integrated into a single skills ontology. At this stage the skills trees reached a maturity that the combination of them caused no major changes for the single skills trees.

4.3 Refinement

During the **refinement** phase we formalized and integrated the semi-formal ontology descriptions into a single coherent skills ontology. An important aspect during the formalization was (i) to give the skills proper names that uniquely identify each skill and (ii) to decide on the hierarchical structure of the skills. We discussed two different approaches for the hierarchical ordering: we discovered that categorization of skills is typically not based on an `is-a`-taxonomy, but on a much weaker HASSUBTOPIC relationship that has implications for the inheritance of attached relations and attributes. However, for our first prototype this distinction made no difference due to missing cross-taxonomical relationships. But, according to [7], subsumption provided by `is-a` taxonomies is often misused and a later formal evaluation of the skills ontology according to the proposed OntoClean methodology possibly would have resulted in a change of the ontology.

In a second refinement cycle we added one more relation type, an "associative relation" between concepts. They express relations outside the hierarchic skills tree, *e.g.* a relation between "HTML" and "JSP", which occur not in the same tree, but correspond with each other, because they are based on the same content. "HTML" is in the tree "markup languages", while the tree "scripting languages" contains "JSP". This is based on the basic characteristics and the history of both concepts, which changed over time. But in reality they have a close relationship, which can be expressed with the associative relation.

The other task in this phase was to integrate the three skills ontologies into one skills ontology and eliminate inconsistencies in the domain ontology parts and between them. Because the domain ontologies were developed separately, the merger of them caused some overlaps, which had to be resolved. This happened for example in the computer science part of the skills trees, where the departments IT and private insurance have the same concepts like "Trofit" (which is a Swiss Life specific application). Both departments use this concept, but each uses a different view. The IT from the development and the private insurance from the users view. Additionally the personal skills of any employee are graded according to a generic scale of four levels: basic knowledge, practical experience, competency, and top specialist. The employees will grade their own skills themselves. As known from personal contacts to other companies (*e.g.* Credit Suisse, ABB and IBM), such an approach proved to produce highly reliable information.

As a **result** at the end of the refinement phase the "target skills ontology" consisted of about 700 concepts, which could be used by the employees to express their skill profile.

4.4 Application & Evolution

The **evaluation** of the prototype and the underlying ontology was unfortunately skipped due to internal restructuring at Swiss Life which led to a closing down of the whole case study.

Still, we considered the following aspects for the **evolution** of our skills management application: The competencies needed from employees are a moving target. Therefore the ontologies need to be constantly evaluated and maintained by experts from the human resource department. New skills might be suggested by the experts themselves, but mainly by employees. Suggestions include both, the new skill itself as well as the position in the skills tree where it should be placed. While employees are suggesting only new skills, the experts decide which skills should change in name and/or position in the skills tree and, additionally, decide which skill will be deleted. This was seen as necessary to keep the ontology consistent and to avoid that *e.g.* similar if not the

elling.

same concept appear even in the same branch. For each ontology (and domain) there should exist a designated ontology manager who decides if and how the suggested skill is integrated.

5. CONCLUSION

The described methodology was developed and applied in the On-To-Knowledge project. One of the core contributions of the methodology that could not be shown here is the linkage of available tool support with case studies by showing when and how to use tools during the process of developing and running ontology based applications in the case studies (cf. [18]).

Lessons learned during setting up and employing the methodology in the On-To-Knowledge case studies include: (i) different processes drive KM projects, but "Human Issues" might dominate other ones (as already outlined by Davenport [3]), (ii) guidelines for domain experts in industrial contexts have to be pragmatic, (iii) collaborative ontology engineering requires physical presence *and* advanced tool support and (iv) brainstorming is very helpful for early stages of ontology engineering, especially for domain experts not familiar with modelling (more details on be found e.g. in [16, 17]).

In this paper we have shown a process oriented methodology for introducing and maintaining ontology based knowledge management systems. Core to the methodology are Knowledge Processes and Knowledge Meta Processes. While Knowledge Meta Processes support the setting up of an ontology based application, Knowledge Processes support its usage. Still, there are many open issues to solve, *e.g.* how to handle the evolution of ontologies on a technical level.

6. ACKNOWLEDGMENTS

The research presented in this paper greatly benefits from contributions of our colleagues at the Institute AIFB, University of Karlsruhe, the closely related company Ontoprise GmbH and our project partner SwissLife. We thank especially Hans-Peter Schnurr (now Ontoprise GmbH) and Hans Akkermans (VU Amsterdam) for their seminal work while setting up the baseline version of this methodology. Part of this work has been financed by the EU in the project IST-1999-10132 On-To-Knowledge.

7. REFERENCES

[1] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, and M. Sintek. Toward a technology for organizational memories. *IEEE Intelligent Systems*, pages 40–48, May / June 1998.

[2] T. Buzan. *Use your head.* BBC Books, 1974.

[3] T. Davenport and L. Prusak. *Working Knowledge - how organisations manage what they know.* Havard Business School Press, Boston, Massachusetts, 1998.

[4] J. Davies, D. Fensel, and F. van Harmelen, editors. *Towards the Semantic Web – Ontology-Driven Knowledge Management.* Wiley, 2002.

[5] R. Dieng, O. Corby, A. Giboin, and M. Ribiere. Methods and tools for corporate knowledge management. *Int. Journal of Human-Computer Studies*, 51(3):567–598, 1999.

[6] T. Gruber. Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, (43):907–928, 1995.

[7] N. Guarino and C. Welty. Evaluating ontological decisions with OntoClean. *Communications of the ACM*, 45(2):61–65, February 2002.

[8] I. Horrocks and J. A. Hendler, editors. *The Semantic Web - ISWC 2002, First International Semantic Web Conference, Sardinia, Italy, June 9-12, 2002, Proceedings*, volume 2342 of *Lecture Notes in Computer Science*. Springer, 2002.

[9] T. Lau and Y. Sure. Introducing ontology-based skills management at a large insurance company. In *Proceedings of the Modellierung 2002*, Tutzing, Germany, March 2002.

[10] M. F. Lopez, A. Gomez-Perez, J. P. Sierra, and A. P. Sierra. Building a chemical ontology using Methontology and the Ontology Design Environment. *Intelligent Systems*, 14(1), January/February 1999.

[11] D. O'Leary. Using AI in knowledge management: Knowledge bases and ontologies. *IEEE Intelligent Systems*, 13(3):34–39, May/June 1998.

[12] G. Probst, K. Romhardt, and S. Raub. *Managing Knowledge.* J. Wiley and Sons, 1999.

[13] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde, and B. Wielinga. *Knowledge Engineering and Management — The CommonKADS Methodology.* The MIT Press, Cambridge, Massachusetts; London, England, 1999.

[14] S. Staab, H.-P. Schnurr, R. Studer, and Y. Sure. Knowledge processes and ontologies. *IEEE Intelligent Systems, Special Issue on Knowledge Management*, 16(1), January/Febrary 2001.

[15] Stojanovic, L., Maedche, A., Motik, B., & Stojanovic, N. (2002). User-driven ontology evolution management. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, LNAI 2473, pages 285–300, Siguenza, Spain.

[16] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke. OntoEdit: Collaborative ontology development for the semantic web. In *[8]*, pages 221–235, 2002.

[17] Y. Sure, S. Staab, and J. Angele. OntoEdit: Guiding ontology development by methodology and inferencing. In *Proceedings of the International Conference on Ontologies, Databases and Applications of SEmantics ODBASE 2002*, University of California, Irvine, USA, 2002. Springer, LNCS.

[18] Y. Sure and R. Studer. On-To-Knowledge Methodology — final version. On-To-Knowledge deliverable D-18, Institute AIFB, University of Karlsruhe, 2002.