Introducing Ontology-based Skills Management at a large Insurance Company

Thorsten Lau[#] and York Sure[+]

[#]Swiss Life,
CC/ITRD
P.O. Box
CH-8022 Zürich
+41 (0) 1 284 4870
thorsten.lau@swisslife.ch

[+]Institut AIFB
Universität Karlsruhe
Postfach
D-76128 Karlsruhe
+49 (0) 721 608 6592
sure@aifb.uni-karlsruhe.de

Abstract Knowledge Management (KM) has become an important success factor for enterprises in virtually all areas during the last decade. IT-supported KM solutions are built around some kind of organizational memory that is structured according to an ontology. We present a methodology for application-driven development of ontologies that is instantiated by a case study, viz. the introduction of an ontology-based skills management system at Swiss Life and the lessons learned from the utilisation of the methodology.

Keywords: Ontology, Skills Management

1 Introduction

Knowledge Management (KM) has become an important success factor for enterprises in virtually all areas during the last decade — to name but a few, one might think of human resource management, enterprise organization and enterprise culture. Information technology (IT) plays a crucial role in knowledge management, *e.g.* by operationalizing knowledge management processes in daily life.

IT-supported KM solutions are built around some kind of organizational memory [KA97] 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 [Die99]. 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 modelling problems by providing a formal conceptualisation of a particular domain that is shared by a group of people in an organization [Gru95][Lea98].

Typically, an ontology consists of named concepts which are organized in an *is-a* hierarchy (subconcept *is-a* superconcept), attributes attached to these concepts, typed relations between concepts and instances of these concepts. In our scenario we rely on state-of-the art representation languages, viz. RDF(S) [RDFS] providing a basic ontology language and it's extension DAML+OIL [DAM01] adding additional logical structures.

At the moment, the development of ontologies for knowledge management come to a point where companies adopt the ideas of ontologies. Therefore it is necessary to offer a

methodology for the development, employment and maintenance of ontologies. As part of the EU project On-To-Knowledge[1] a methodology for ontologies was developed and tested. The methodology was developed in a joint effort of the methodology provider, i.e. the Institute AIFB, and the case study partners, e.g. Swiss Life, and provides a generic process model for the ontology development, which is instantiated by the case studies. Swiss Life provides as case study a skills management system, which has to deal with the modelling of an ontology from a broader kind of view, encompassing a large ontology, a fuzzy domain and different usage scenarios.

In this paper we will describe briefly the methodology and the usage of it developing an ontology for a skills management system at Swiss Life. We describe the generic methodology in chapter 2, the utilisation of the methodology in the skills management in chapter 3 and the resulting lessons learned in chapter 4. Chapter 5 contains related work and a summary.

2 Methodology for ontology development

Ontologies are a core element of the On-To-Knowledge project. Our methodology (cf. [Sta01]) therefore has a strong focus on developing ontologies. In contrast to well-known methodologies for ontology development, which mostly restrict their attention within the ontology itself (e.g. [UK95], [Lop99]), our approach focuses on the application-driven development of ontologies during the introduction of ontology-based knowledge management systems. We cover aspects from the early stages of setting up a knowledge management project to the final roll out of the ontology-based knowledge management system. The steps of our methodology are sketched in Figure 1, we will now describe each step in detail.

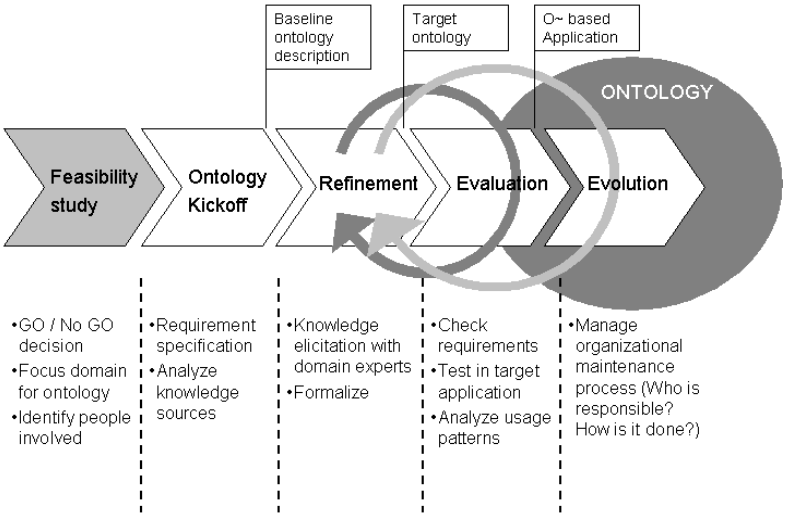

Figure 1: Steps of the OTK Methodology

[1] More Information can be found at: http://www.ontoknowledge.org

2.1 Feasibility study

Any knowledge management system may only function satisfactorily if it is properly integrated into the organization in which it is operational. Many factors other than technology determine success or failure of such a system. To analyse these factors, one must initially perform a feasibility study to first identify problem / opportunity areas and potential solutions, and second, to put them into a wider organizational perspective. The feasibility study serves as a decision support for economical and technical project feasibility, in order to select the most promising focus area, i.e. the domain for the ontology-based system to be developed. We rely on the approach for carrying out a feasibility study that is described by the CommonKADS methodology [Sch99]. It should be carried out before actually developing ontologies and serves as a basis for the kick-off phase. Besides the domain of the system it helps to identify the people involved in setting up and using the system, viz. the domain experts, users and supporters of a system.

2.2 Kick-off phase

The first step to actually engineer ontologies is to capture requirements in an Ontology Requirements Specification Document ("ORSD") describing what an ontology should support and sketching the planned area of the ontology application. It should guide an ontology engineer to decide about inclusion, exclusion and the hierarchical structure of concepts in the ontology. In this early stage one should look for already developed and potentially reusable ontologies. In summary, it should clearly describe the information shown in Table 1.
Through analysis of the available knowledge sources a "baseline ontology" is gathered, i.e. a draft version containing few but seminal elements of an ontology. Typically the most important concepts and relations are identified on an informal level. A very important knowledge source (also for the later phases) are domain experts. In our case study we heavily relied on domain experts for the ontology development (see next section).

ORSD1.	Domain and goal of the ontology
ORSD2.	Design guidelines to ensure a consistent development (e.g. naming conventions)
ORSD3.	Available knowledge sources (e.g. domain experts, reusable ontologies, organization charts, dictionaries, index lists, db-schemas etc.)
ORSD4.	Potential users and use cases
ORSD5.	Applications supported by the ontology

Table 1: Content of the ORSD

2.3 Refinement phase

The goal of the refinement phase is to produce a mature and application-oriented "target ontology" according to the specification given by the kick-off phase. This phase is divided into different sub-phases shown in Table 2.
This phase is closely linked to the evaluation phase. If the analysis of the ontology in the evaluation phase shows gaps or misconceptions, the ontology engineer takes these re-

sults as input for the refinement phase. It might be necessary to perform several iterative steps.

R1.	A knowledge elicitation process with domain experts based on the initial input from the kick-off phase. This serves as input for further expansion of the baseline ontology. Typically axioms are identified and modelled in this phase. This is closely linked to the next step - the effects of axioms might depend on the selection of the representation language.
R2.	A formalization phase to transfer the ontology into the "target ontology" expressed in formal representation languages like DAML+OIL [DAM01], The representation language is chosen according to the specific requirements of the envisaged application.

Table 2: Two sub-phases of the refinement phase

2.4 Evaluation phase

The evaluation phase serves as a proof for the usefulness of developed ontologies and their associated software environment. In a first step, the ontology engineer checks, whether the target ontology fulfils the ontology requirements specification document and whether the ontology supports or "answers" the competency questions analysed in the kick-off phase of the project. In a second step, the ontology is tested in the target application environment. Feedback from beta users may be a valuable input for further refinement of the ontology.
A valuable input may be as well the usage patterns of the ontology. The prototype system has to track the ways users navigate or search for concepts and relations. With such an "ontology log file analysis" we may trace what areas of the ontology are often "used" and others which were not navigated. As mentioned before, this phase is closely linked to the refinement phase and an ontology engineer may need to perform several cycles until the target ontology reaches the envisaged level - the roll out of the target ontology embedded into the ontology-based application finishes the evaluation phase.

2.5 Maintenance phase

In the real world things are changing - and so do the specifications for ontologies. To reflect these changes ontologies have to be maintained frequently like other parts of software, too. We stress that the maintenance of ontologies is primarily an organizational process. There must be strict rules for the update-insert-delete processes within ontologies. Most important is to clarify who is responsible for maintenance and how it is performed. E.g. is a single person or a consortium responsible for the maintenance process? In which time intervals is the ontology maintained? We recommend that the ontology engineer gathers changes to the ontology and initiates the switch-over to a new version of the ontology after thoroughly testing possible effects to the application, viz. performing additional cyclic refinement and evaluation phases. Similar to the refinement phase, feedback from users may be a valuable input for identifying the changes needed. Maintenance should accompany and applications and their embedded ontologies as long as they are on duty.

3 Case Study Skills Management

Swiss Life as the case study partner in the On-To-Knowledge project identified Skills Management as one important part in the vision of an Organizational Memory (OM) [KA97], which integrates knowledge sources of different kinds. The vision is to build an intranet-based portal with a single user access point to all knowledge sources within the company. Skills management is one of these sources and Swiss Life implemented a prototype to evaluate the advantages of ontologies.
The aim of this prototype is an integrated application system in the area of skills management for employees within Swiss Life, which makes skills of employees explicit. Two aspects are covered by the system: on the one hand allow explicit skills as yellow pages for every employee for an advanced expert search within the intranet, which can be used by all employees. On the other hand one might explore his future career path by matching his current skill profile vs. pre-defined job profiles, also called gap analysis.

Skills management covers numerous aspects of the human resources management, e.g. the management of CVs, certificates, recruitment profiles, project description, etc. The skills management system comprises four components:

- Description and management of skills
- Templates that employees can use to build personalised intranet home pages
- Intranet-Homepages
- Search facility to search the home pages for specific skills

The prototype enables any employee to produce in a simple way an individual Intranet home page including information of personal skills. Any home page will contain (see Figure 2):

- general information, such as functional unit, room, or telephone number (organisational data),
- other public descriptions, such as education, current position,
- details of personal skills, and
- a section that employees are free to fill in with personal interests, hobbies and categories, such as 'I am a member of the professional associations...', or 'I am familiar with the following specialised literature...'

To ensure that all knowledge sources are used in the same way, ontologies are used as a common mean of interchange to face two major challenges: First, being an international company located in Switzerland, Swiss Life has internally four official languages, viz. German, English, French and Italian. Second, there exist several spellings of same concepts, e.g. “WinWord” vs. “MS Word”. To tackle these problems, ontologies offer mappings to different languages and allow for synonyms.

To develop the underlying ontologies for the skills management system, we relied on the On-To-Knowledge methodology (see previous Section). The actual process of instantiating the methodology for the case study is described in the following.

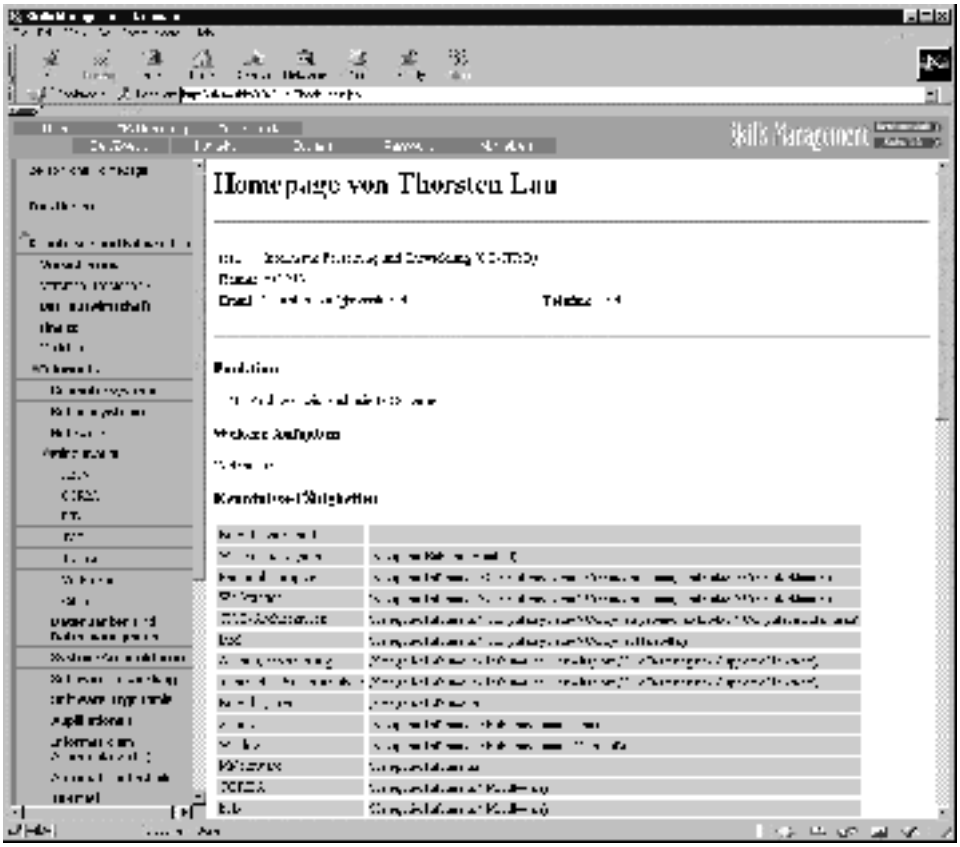

Figure 2: Homepage of an employee

3.1 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 [LNR00] of the existing skills management environment and evaluated the needs for a new skills management system. As actors and stakeholders for the skills management we identified mainly the human resources department as well as the management level of all other departments. 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.

3.2 Kick-off phase

The departments private insurance, human resources and IT as three different domains were the starting point for an initial prototype. Therefore the task was to develop a skills ontology for the departments containing three trees, viz. for each department one. These three trees should be combined under one root with having cross-links in between. This

root node is the abstract concept *skills* (which means in German "Kenntnisse/Fähigkeiten") and is the starting point to navigate through the skills tree from the top.
Additionally for the gap analysis two ontologies are needed with the functions and education of the employees. The function ontology contains job descriptions available at Swiss Life with the therefore needed skills. In the education ontology all certificates, diploma, etc. were included.
The aim of developing the education and functions ontologies is to know the educational background of an employee, which can be matched to the requirements of the function the employee has or want to have. In the latter case the gap analysis is a means for the further qualification so that the employee is able to change the function.
During the kick-off phase two workshops with the domain experts 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 für 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 easy understandable model of the skills. The domain experts and potential users felt very uncomfortable with the extracted structures and 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 initial lexicon or terminology). During this stage a lot of concept islands were developed, which were isolated bunchs 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 to a single tree. This was a more difficult part as 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. These skills trees would effect confusion on the users side, because the integration of the concepts looks not natural for the user.
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 is dependent of the domain and should be determined by the domain expert.
As result of the kick-off phase we obtained three skills trees, which are ready to be integrated into a single skills ontology. The skills trees reached at this stage a maturity that the combination of them causes no major changes for the single skills trees.

3.3 Refinement phase

During the refinement phase, 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 "mark-up 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 trees 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 from 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. In other companies (e.g. Credit Suisse, ABB and IBM), such an approach proved to produce highly reliable information. At the end of the refinement phase the skills trees consists of about 700 concepts.

3.4 Evaluation and maintenance phase

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. They also decide which skill will be deleted. For each ontology (and domain) there exists a designated ontology manager who decides if the suggested skill is integrated.

4 Lessons learned

During the development of the skills management system we discovered several "lessons learned" [Nov01]. They reflect differences between the methodology and the requirements of the "real life" emerged during all phases of the ontology modelling process. In this section we describe some resulting adaptations of the methodology to the special conditions of the skills management case study.

- Manual vs. automatic ontology development:

We discovered that a decision for a purely manual or (semi-) automatic ontology development approach is dependent on the domain type. Domains use vocabularies, which

range from structured to unstructured. A structured vocabulary is common to all domain users and contains only a small set of synonyms and ambiguous terms, while unstructured vocabularies contain many words, which are used by the domain users in different ways and meanings. The skills management domain is a more unstructured, a fuzzy one, because the vocabulary is shared by a lot of different people and departments with different backgrounds. For example, the private insurance department of Swiss Life has a completely different view and understanding on the terms "insurance premiums", "customer" and "policyholder" as the group insurance department, which is based on different legislation. Therefore a common vocabulary does not exist and has to be agreed between the departments manually. This is foremost reason why the automatic ontology development approach for the skills ontology development failed. The information extraction tools typically are not able to match different words of a concept (synonyms) to a single term. It should be in mind that this structuring process takes much more time than the automatic approach based on the background to integrate different views of the departments into one common view, which tends to be complicated task depending on the people and the similarities between the departments.
The lesson learned of this experience is that the domain has to be structured in front of the ontology modelling process. This is needed for an automatic ontology development as well as it is needed for unifying the different vocabularies of an unstructured, fuzzy domain. In the latter case the structuring process is manual work, which has to be done by domain experts like we have done for the skills management system. A possible approach to speed up the development process is to build only a common top-level ontology for all departments and keep the individual parts of the departments. The disadvantage of this approach is that the common understanding of the ontology can be lost. Each department still uses their vocabulary and the match between these is not documented. Additionally, some parts of the ontology can hold similar structures without noticing it, which is based on the separated and non coordinated development of the single parts.

- Abstraction levels and size:

Several people were involved in creating the first draft versions of the ontology during early stages in the kick-off phase of the modelling process. Due to different backgrounds and mind models of the developers all of them had different abstraction levels. This made it difficult to merge them into a single coherent ontology. Typical problems rose from differing numbers of branches per node and more general from how deep and wide the ontology is at all. Many iterations in the ontology development process were necessary to align the ontologies and finally to merge them together. We stress that the ontology modelling process should start with a definition of the abstraction level, which is strongly dependent on the usage of the ontology. In our case, we had to go down to a more precise level during the modelling process due to the high number of employees and the need to differentiate them for the intended project staffing functionality.
First feedback from test users of the skills management system and the underlying ontology showed that the skills trees are too large for browsing. Most users prefer shallow trees with a list of concepts on each node on a more abstract level. The reasons for this are the high time consuming usage of the deep ontology concepts and the amount of possible concepts, which can be chosen. The latter reason is based on the integrated skills ontology of all domains or departments. A suggestion from the test users was to define user views on the ontology depending on the department where the user comes from. This would lead to much smaller skills trees and more usable ones in the according

departments. Employees with broader skills can change from the department/domain view to the interdisciplinary view on the whole ontology. Though views are common to the database community, no concept for defining and using views on ontologies has been developed so far.
As a summary, it is a challenging task to find the right balance between the requirements from the user of the skills management system and the ontology user. While the first wants to have a large ontology, the latter one feels comfortable with a small ontology.

- Concepts vs. instances:

A typical problem area known from expert systems is the boarder between concepts and instances. Though most knowledge engineers with a practical background prefer not to distinguish between concepts and instances during the modelling process at all, our underlying architecture made it necessary to make this distinction upfront. As mentioned above it is difficult to distinguish between a leaf node of the concept hierarchy and the belonging instances. Our lesson learned in this area is that it is a good rule of thumb to count the possible instances and decide if this will be enough to justify a concept or not. In the case of "Java" and "jdk1.3" we assumed that not so many employees will use "jdk1.3" so that our decision was to take "Java" as leaf node. In general it should be discussed close together with the definition of the abstraction and depth/width of the ontology, which builds a trade-off. Though there exists the simple heuristic to take every leaf node as an instance, this seems not to be correct for all use cases.

- De-contextualization of concepts:

Continuing the ontology modelling, a further difficulty emerged with the "de-contextualisation" of ontology concepts. We had the problem that the ontology developers used concept names, such as "basics" in the ontology. For a human reader the meaning of such a concept is only comprehensible if its super-concept is known. When browsing the ontology this is no problem, because the super-concept is visible. But when a user selects this element as a skill then just "basics" occurs in his homepage, i.e. de-contextualiziation of this concept is made. The same problem occured with "informatics" which occurs as a sub-concept of "skill", "function" and "education" resulting in three different meanings of the concept ("informatics-skills", "informatics-function" and "informatics-education").
It is no solution to force the ontology developers to use concept names that include the context of this concept. This would result in very long concept names. Furthermore, the modellers often forget about this problem and it is very hard to explain them why the concept name has to show the whole context. Therefore, we decided to "re-contextualizse" the concept in the homepage by showing the path to the root concept of the ontology. So far we made good experiences but it is too early to say if this solution will hold for all cases.

5 Related work

There exist various Skill Management solutions within ERP systems from companies like SAP and Meta4, that typically do not publish scientifically about their systems. The usage of ontologies for Skill Management systems is new and although we only slightly extended typical database functionalities so far, especially multilingual mappings, syno-

nyms and the integration of various knowledge sources have already shown the usefulness of our approach [Nov01].
The system described in [SMS00] gave first ideas how to implement an ontology-based skills management system. Like proposed we integrated different legacy systems from Swiss Life in the Skills Management area and put the ontology on top. Still missing in our current prototype is an advanced reasoning service which will be included in the next version. There exist only few other methodologies for building ontologies. We present two major methodologies that influenced our own scenario.
The Skeletal Methodology is based on the experience of building the Enterprise Ontology [UK95], which includes a set of ontologies for enterprise modelling. The guidelines for developing ontologies start with identifying the purpose of an ontology and then concentrate on the building of ontologies which is broken down into the steps ontology capture, coding, evaluation and documentation. A disadvantage of this methodology is that it does not precisely describe the techniques for performing the different activities. For example, it remains unclear, how the key concepts and relationships should be acquired. Only a very vague guideline is given.
The Methontology framework from [Lop99] includes four steps. (1) The identification of the ontology development process, which refers to which tasks (planning, control, specification, knowledge acquisition, conceptualization, integration, implementation, evaluation, documentation, configuration management) one should carry out, when building ontologies. (2) The identification of stages through which an ontology passes during its lifetime. (3) The steps to be taken to perform each activity, supporting techniques and evaluation steps. (4) Setting up an ontology requirements specification document (ORSD) to capture requirements for an ontology similar to a software specification. The methodology offers detailed support in development-oriented activities except formalization and maintenance and describes project management activities. We adopted the general idea of an ontology requirements specification document (ORSD), but modified and extended the presented version by our own needs.

6 Conclusion

We presented a methodology for developing application-driven ontologies. The process model contains five major steps, viz. a feasibility study, kick off, refinement, evaluation and maintenance. The methodology is instantiated by a case study at Swiss Life that deals with the introduction of ontology-based skills management. We described the ontology modelling for the system according to the methodology and gave some additionally lessons learned that reflect the differences between the general methodology and real life conditions in a company.

We implemented a prototypical system that already is able to support employees in their daily life. It is currently used by over 200 users and is still growing. For the future we plan to expand the system by more advanced reasoning services, e.g. we will expand the skills tree by adding also cross-taxonomic relationships: people having skills in “Java” typically have also some capabilities in related areas like “JSP”. Thereby we will rank results from queries about persons having a specific skill according to the strength of the relationship between two related skills.

Acknowledgements: First, we thank Bernd Novotny who was responsible for the Skills Management project until July 2001 and realised the prototype and first evaluation of the system. Furthermore we want to thank Urs Gisler, Valentin Schöb and Patrick Shann for the ontology modelling and all our colleagues from the project without whom the research would not have been possible. The research presented in this paper has been partially funded by EU in the IST-1999-10132 project On-To-Knowledge and the Swiss Government under the contract-no. "BBW Nr. 99.0174".

References

[DAM01] Reference description of the DAML+OIL ontology markup language, http://www.daml.org/2001/03/reference.html , March 2001.

[Die99] 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.

[Gru95] Gruber, T. (1995). Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies, (43):907–928.

[KA97] O. Kuehn and A. Abecker. Corporate memories for knowledge memories in industrial practice: Prospects and challenges. Journal of Universal Computer Science, 3(8), August 1997.

[Lea98] D. O'Leary. Using AI in knowledge management: Knowledge bases and ontologies. IEEE Intelligent Systems, 13(3):34– 39, May/June 1998.

[LNR00] T. Lau, B. Novotny and U. Reimer. Case Studies on Organizational Memory. Deliverable 19, EU-Project On-To-Knowledge, September 2000.

[Lop99] M. F. López, A. Gómez-Pérez, J. P. Sierra, and A. P. Sierra. Building a Chemical Ontology Using Methontology and the Ontology Design Environment. Intelligent Systems & their applications, Vol. 14, No. 1, January/February 1999.

[Nov01] B. Novotny, T. Lau, J. Reich, U. Reimer. Organizational Memory - Evaluation of the Case Study Prototypes. Deliverable 21, EU-Project On-To-Knowledge, July 2001.

[RDFS] W3C. RDF Schema Specification. http://www.w3.org/TR/PR-rdf-schema/, 1999

[Sch99] 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.

[SMS00] Y. Sure, A. Maedche, and S. Staab: Leveraging Corporate Skill Knowledge - From ProPer to OntoProPer. In: D. Mahling & U. Reimer. Proceedings of the Third International Conference on Practical Aspects of Knowledge Management (PAKM 2000), 30 - 31 October 2000, Basel, Switzerland.

[Sta01] S. Staab, H.-P. Schnurr, R. Studer, and Y. Sure: „Knowledge Processes and Ontologies". In: IEEE Intelligent Systems 16(1), January/February 2001, Special Issue on Knowledge Management.

[UK95] M. Uschold and M. King. Towards a Methodology for Building Ontologies. In Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95, Montreal, Canada, 1995