# On-To-Knowledge Methodology — Baseline Version

**Hans-Peter Schnurr, York Sure, Rudi Studer (University of Karlsruhe)**
**Hans Akkermans (Vrije Universiteit Amsterdam)**

**Executive Summary.**
On-To-Knowledge EU-IST-1999-10132 Project Deliverable D15
(WP5, due month 8), 31 August 2000.

| | |
|---|---|
| **Document Id.** | OTK/2000/15/v1.0 |
| **Project** | On-To-Knowledge |
| **Date** | 31 August 2000 |
| **Distribution** | consortium, EC |

Copyright © 2000 On-To-Knowledge consortium

## On-To-Knowledge Consortium

This document is part of a research project partially funded by the IST Programme of the Commission of the European Communities as project number IST-1999-10132. The partners in this project are: Vrije Universiteit Amsterdam VUA (coordinator, NL), University of Karlsruhe (Germany), Schweizerische Lebensversicherungs- und Rentenanstalt/Swiss Life (Switzerland), British Telecommunications plc (UK), CognIT a.s. (Norway), EnerSearch AB (Sweden), AIdministrator Nederland BV (NL).

**Vrije Universiteit Amsterdam (VUA)**
Division of Mathematics and Informatics W&I
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands
Tel: +31 20 4447718, Fax: +31 20 872 27 22
Contactperson: Dieter Fensel
E-mail: dieter@cs.vu.nl

**University of Karlsruhe**
Institute AIFB
Kaiserstr. 12
D-76128 Karlsruhe
Germany
Tel: +49 721 608392, Fax: +49 721 693717
Contactperson: R. Studer
E-mail: studer@aifb.uni-karlsruhe.de

**Schweizerische Lebensversicherungs- und Rentenanstalt / Swiss Life**
Swiss Life Information Systems Research Group
General Guisan-Quai 40
8022 Zürich
Switzerland
Tel: +41 1 284 4061, Fax: +41 1 284 6913
Contactperson: Ulrich Reimer
E-mail: Ulrich.Reimer@swisslife.ch

**British Telecommunications plc**
BT Adastral Park
Martlesham Heath
IP5 3RE Ipswich
United Kingdom
Tel: +44 1473 605536
Fax: +44 1473 642459
Contactperson: John Davies
E-mail: John.nj.Davies@bt.com

**CognIT a.s**
Busterudgt 1.
N-1754 Halden
Norway
Tel: +47 69 1770 44, Fax: +47 669 006 12
Contactperson: Bernt. A.Bremdal
E-mail: bernt@cognit.no

**EnerSearch AB**
Carl Gustafsväg 1
SE 205 09 Malmö
Sweden
Tel: +46 40 25 58 25; Fax: +46 40 611 51 84
Contactperson: Hans Ottosson, Fredrik Ygge
E-mail: hans.ottosson,fredrik.ygge@enersearch.se

**AIdministrator Nederland BV**
Julianaplein 14B
3817 CS Amersfoort
The Netherlands
Tel: +31 33 4659987, Fax: +31 33 4659987
Contactperson: Jos van der Meer
E-mail: Jos.van.der.Meer@aidministrator.nl

# Contents

# Chapter 1

# Introduction

The On-To-Knowledge project applies ontologies to electronically available information to improve the quality of knowledge management in large and distributed organisations. This enables sophisticated automatic support for acquiring, maintaining, and accessing information. For this we will develop a methodology and tools for intelligent access to large volumes of semistructured and textual information sources in intra-, extra-, and internet-based environments. The methodology provides guidelines for introducing knowledge management concepts and tools into enterprises, helping knowledge providers and seekers to present their knowledge efficiently and effectively. The methodology includes the identification of goals that should be achieved by knowledge management tools and is based on an analysis of business processes and the different roles knowledge workers play in organisations. To get a baseline version of the methodology, we elaborate on the aspects and issues as depicted in Figure 1.1.

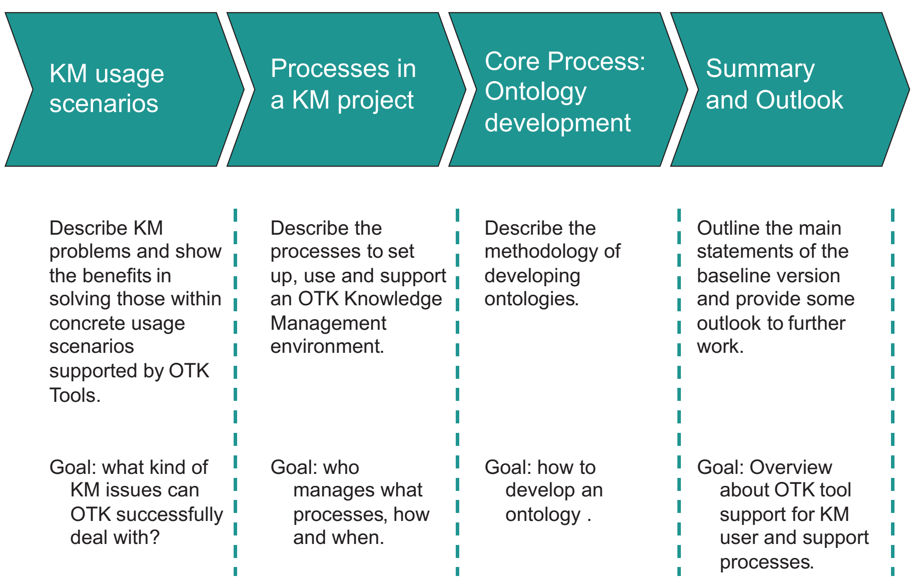


Figure 1.1: Aspects and issues of the baseline OnToKnowledge Methodology

In Section 2 we continue with the description of knowledge management problems in concrete usage scenarios supported by OnToKnowledge (OTK) Tools. In Section 3, we describe the processes to set up, use and support an OnToKnowledge KM environment. This includes a feasability study, several primary end user related processes and secondary processes to set up, support and maintain the system. The main focus of the OnToKnowledge project is to apply ontologies to electronically available information to improve the quality of knowledge management. In Section 4, we describe the development of ontologies. Section 5 summarizes the baseline version of the methodology and gives an overview, how the OnToKnowledge tools support the primary and secondary KM processes. An outlook about the further work on evaluation and gradual expansion of the baseline version is given in Section 6.

# Chapter 2

# The KM challenge

## 2.1 Knowledge access problems

KM systems must provide a simple and effective access to knowledge with low costs and without barriers. Existing keyword-based information retrieval techniques clearly fail on these requirements. Due to that fact, systems developed on top of these techniques like *e.g.* current Document-Management-Systems have severe drawbacks (*cf.* (van Harmelen & Fensel, 1999)):

- *Searching information*: Existing keyword-based search also retrieves irrelevant information that uses a certain term in a different meaning, and misses information when different terms with the same meaning about the desired content are used.
- *Extracting information*: Currently, human browsing and reading is required to extract relevant information from information sources since automatic agents do not possess the commonsense and domain knowledge required to extract such information from textual representations, and they fail to integrate information spread over different sources.
- *Maintaining* weakly structured text sources is a difficult and time-consuming activity when such sources become large. Keeping such collections consistent, correct, and up-to-date requires mechanized representations of semantics that help to detect anomalies.
- *Automatic document generation* (Perkowitz & Etzioni, 1997) would enable adaptive websites that are dynamically reconfigured according to user profiles or other aspects of relevance. Generation of semi-structured information presentations from semi-structured data requires a machine-accessible representation of the semantics of these information sources.

## 2.2 Usage scenarios

We will now present a short overview of major challenges occuring in usage scenarios of the OnToKnowledge project. All these problems are described from a user s point of view which we use as a starting point to explore expected capabilities of the OTK toolset. In the following Section 3 we will make a transfer from the described usage scenarios to use cases relevant in KM projects and go into more detail by identifying stakeholders and their interaction with the OnToKnowledge toolset.

### 2.2.1 Sharing knowledge

Discussion forums became very popular for exchanging knowledge related to a specific topic with other people. Companies implemented different kinds of discussion forums and in the beginning users took actively part in sharing their knowledge with each other. But usage on a regular basis is usually restricted by the following problems:

- Motivation
- Overhead work
- None or weak personalization features available

Without having a strong personal interest and motivation, most users quit sharing their knowledge very quickly. Also, users are not willing — or simply not capable due to high pressure project work — to invest time for knowledge sharing without having direct benefits and therefore avoid this overhead of work.

### 2.2.2 Querying for knowledge

Corporate intranets and the Internet are populated by millions of documents containing hopefully valuable contents. Currently querying for knowledge in these extensives environments is usually restricted to keyword-based search based on information retrieval techniques (although there exist other approaches which are *e.g.* ontology-based like (Decker et al., 1999), (Luke et al., 1997), (Guarino et al., 1999) or (Domingue & Motta, 2000)). Although widely used, there are some major problems of keyword based search engines like *e.g.* the following:

- Too many results and often not the right ones
- Keyword selection matters
- Generalization / refinement not possible

When looking *e.g.* for pages containing information about the research topics of a researcher called *Feather* (*cf.* (Fensel, 2000)), a typical search engine like *Alta Vista* retrieves a huge amount of pages containing the keyword *Feather*. Many of the retrieved pages will be irrelevant for the user — who has to find his way through the jungle of hits presented.

By using smart combinations of keyword for retrieval, one may restrict given hits to a more manageable amount with sometimes even more relevant hits. The important part is: *smart* — even experienced users need sometimes several tries to find *smart combinations*. To continue the above example: using *Feather+research* as search string offers good chances for finding the right person.

When now looking for all researchers who work *e.g.* within the same community of the researcher *Feather*, a user has several options: trying another *smart combination* — but which one? — or browsing through Feather's Homepage hoping that the researcher provided links to publications from colleagues. From a more general point of view: users are left alone to generalize (or refine) their queries on a conceptual level and to transform this new query again into a keyword based search.

### 2.2.3 Navigating

Additionally to querying for knowledge a user may navigate and browse through documents contained in an intranet or the Internet — as mentioned above, sometimes it's the only way of generalizing or refining queries. But even if navigating is supported *e.g.* by given categorizations of relevant topics, there often occur severe problems while using these categorizations:

- Categorization of users differ from system categories
- None or weak user guidance within categorization
- Missing visualization

Most corporate intranets offer some kind of structured hierarchy for navigating. But system designers points of view do not necessarily match to users points of view which results in confusing views for users.

Some search portals like *Yahoo!* offer categorizations for users. However, when navigating through these categorizations users are weakly or not at all supported by visualization and usually loose orientation quickly.

# Chapter 3

# Use cases in an OTK project

Before we describe the overall picture of an ontology-based KM system provided by the OTK toolset we give a brief introduction to use-case diagrams — those diagrams help clarifying the interactions between stakeholders and processes in an OTK project. Another introductory process to set up a knowledge management system is to proceed a *Feasibility study* based on (Schreiber et al., 1999), *viz.* to identify problem/opportunity areas and potential solutions, and putting them into a wider organizational perspective. The feasibility study serves as a decision support for economical, technical and project feasibility, in order to select the most promising focus area and target solution.

This feasibility study helps to identify (i) stakeholders related to an OTK project divided into *users of the system* and *supporters of the system*, (ii) use cases describing the usage scenarios depicted in Section 2 which we call *user-driven use cases* and (iii) use cases supporting these user-driven use cases which we call *supporting use cases.*

For each use case we present visions as a guideline for possible results of the OnToKnowledge project and finally we provide recommendations on how to support these visions with existing and planned OnToKnowledge tools. These recommendations will be evaluated in the different OTK case studies — and they may change over the time. The next version of our methodology will reflect the changes monitored during the progress of the project.

## 3.1 Brief introduction to use cases

*Use-case diagrams* are notations defined in *UML*, the "Unified Modeling Language" (introduced by (Booch et al., 1998)), which is basically a set of useful notations for (but not restricted to) software design and modeling. Typically they are used during early phases of system development and show which services are provided by a system (*cf.* (Schreiber et al., 1999)). Use cases may interact with so-called *actors* who stand outside the *system.* An actor in a use-case diagram represents a group of actor objects meaning that they are defined on a "class" level. Figure 3.1 depicts the graphical representation of the key elements of use-case diagrams: *use cases*, *system* and *actors* and how they *interact* with use cases. In our context we call these actors *stakeholders* which seems more understandable.

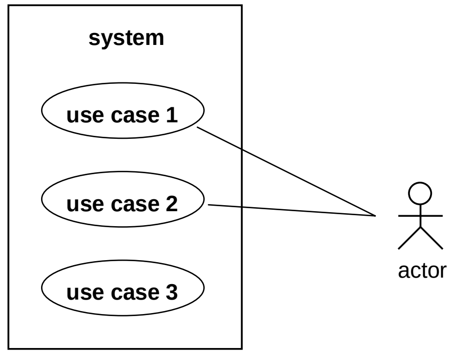


Figure 3.1: Example of an use-case diagram

## 3.2 Feasibility study

Any knowledge management system can function satisfactorily only if it is properly 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 have to proceed a *Feasibility study.* This part of the OnToKnowledge methodology is based on the CommonKADS methodology (*c.f.* (Schreiber et al., 1999)), *viz.* to identify problem/opportunity areas and potential solutions, and putting them into a wider organizational perspective. The feasibility study serves as a decision support for economical, technical and project feasibility, in order to select the most promising focus area and target solution.

For this analysis, the CommonKADS methodology offers three models: the organization, task, and agent model. The process of building these models proceeds in the following steps:

- Carry out a scoping and problem analysis study, consisting of two parts:
  - Identifying problem/opportunity areas and potential solutions, and putting them into a wider organizational perspective.
  - Deciding about economic, technical and project feasibility, in order to select the most promising focus area and target solution.
- Carry out an impacts and improvements study, for the selected target solution, again consisting of two parts:
  - Gathering insights into the interrelationships between the business task, actors involved, and use of knowledge for successful performance, and what improvements may be achieved here.
  - Deciding about organizational measures and task changes, in order to ensure organizational acceptance and integration of a knowledge system solution.

An overview of the process of organizational context modeling is given in Figure 3.2.Building the task, organization and agent model is done by following a series of small steps supported by practical and easy-to-use worksheets and checklists. A detailed description of these steps is given in (Schreiber et al., 1999).

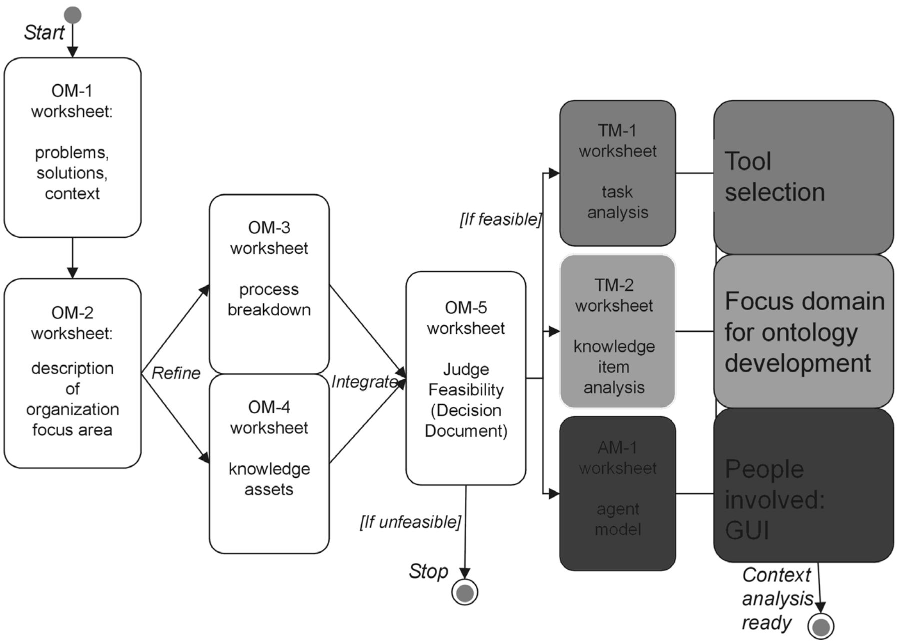


Figure 3.2: Modified CommonKADS steps

For the OnToKnowledge purpose, we modify the output of the feasibility study as indicated in the dark shading in Figure 3.2. The task analysis may serve as a basis for the tool selection, the knowledge item analysis is an input source for the kickoff phase of the ontology development and the agent model indicates the design of an appropriate graphical user interface. The experience and feedback that will be gained from the case studies in the progress of the OnToKnowledge project will be the basis for a smooth integration of the feasibility study and the use cases as described in the following.

## 3.3 Overall picture

As a starting point of drawing an overall picture of a knowledge management project supported by the OTK toolset we take a typical end user *viz.* a *knowledge worker* (*cf.* (Davenport & Prusak, 1998)) who has to *Seek knowledge* to perform her knowledge-intensive tasks. At this she encounters usage scenarios like the ones we described in Section 2: knowledge sharing, navigating and querying for knowledge.

We now map these usage scenarios to a use case diagram as shown in Figure 3.3: *Sharing knowledge (Comm. of know. sharing)*, *Navigating (Nav. /browse KB)* and *Querying for knowledge (Querying KB)*. While describing the use cases we assume a given knowledge base — that might be *e.g.* an XML knowledge repository.

Communities of knowledge sharing usually consist of a bunch of pull services *i.e.* users actively seek information, they actually need to search for the information that they

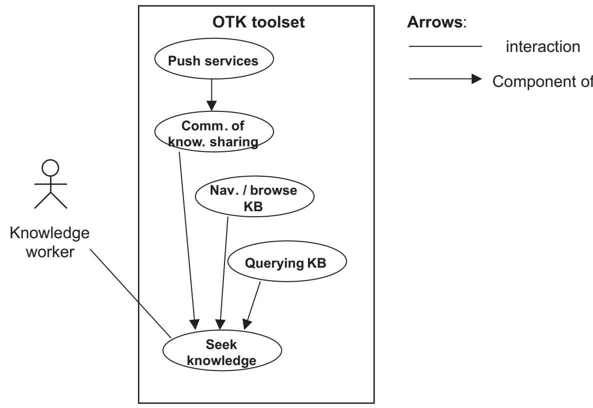


Figure 3.3: Use cases and actors related to the OTK toolset

need to know. In our analysis we found an additional use case, *viz. Push services* to provide knowledge to knowledge workers. Push services distribute and deliver knowledge to knowledge workers after filtering it through highly customized filters (Tiwana, 2000).

For finding knowledge, it first has to be provided, structured and maintained. Several more stakeholders are involved in those service providing functions: *Knowledge engineer*, *Knowledge provider* and *Management* as shown in Figure 3.4. As already mentioned before, we call these stakeholders *supporters of the system*. The management has several important roles in a KM project: it supports the legitimacy of the project and brings in vision that correlates with the overall company-wide vision. The management needs to be thoroughly convinced of the worth of the project (Tiwana, 2000). Therefore it commits the resources needed — and especially assigns people needed for the supporting use cases.

To provide knowledge efficiently and effectively, the OTK toolset helps knowledge providers to present knowledge *viz.* to *Fill Knowledge Base* and subsequently to enrich the filled-in knowledge with *Annotations*.

A well-known means to structure knowledge domains are ontologies, which aim at capturing domain knowledge in a generic way and provide a commonly agreed understanding of a domain, which may be reused and shared across applications and groups (Gomez-Perez & Benjamins, 1999). *Development* and *Maintenance* of ontologies is typically done by *Knowledge engineers*. Due to the fact that ontologies are a core element of the OTK toolset, maintenance of ontologies also triggers maintenance of the annotations as well as maintenance of the knowledge base.

## 3.4 User-driven use cases

An essential element of a successful knowledge management system is the creation of a sound navigational system or framework to make it easy for users to locate the knowledge they seek and to make it easy for users to provide their knowledge they want to share. The user can choose between two general approaches in finding information: one is in *querying the knowledge base*, and the other is in *navigating and browsing the knowledge base*. To

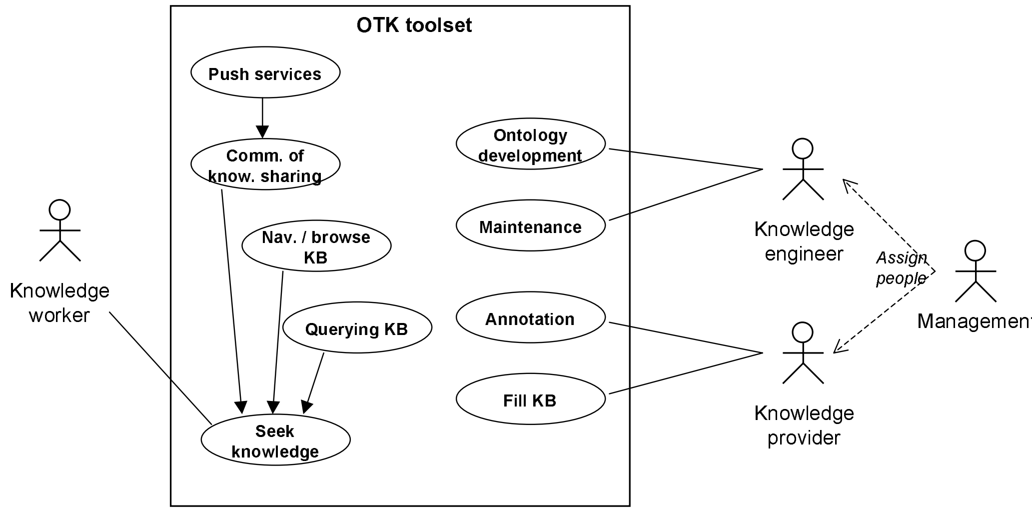


Figure 3.4: Use cases and actors related to the OTK toolset

support the provision and sharing of knowledge, the user has to be guided and supported in a *community of knowledge sharing*. In the following, we will specify a vision how to enable the user-driven use cases within a knowledge management system. This vision describes the desirable functionality of the target system. Subsequently we show possible OnToKnowledge tool support to meet these criteria in the process of the OnToKnowledge project.

### 3.4.1 Querying the knowledge base

#### Vision

As indicated in the usage scenario, queries often follow a simple matching logic. Simple keyword queries are valuable in situations where users have a clear idea of what they are seeking and the information is well-defined. It is, however, limited due to the fact, that simpler keyword searches cannot pick up synonyms and often not even morphological variations, not to mention the context of the query. Our goal is to support users by allowing them to pose semantically rich queries. Therefore, a number of tools and techniques may be of help:

- Answer-seeking users need a helping hand while specifying their scope of interest when posing a query. The visualization of ontological concepts and relations may offer a support to grasp the user s scope of interest. Such a visualization tool helps to select and combine relevant query terms.
- Users may generalize or refine their queries, if a given answer is not satisfactory or the amount of possible answers is too high. A target system should automatically generalize or refine the user s query along the backbone taxonomy of the ontology to deliver an appropriate answer.

- For each specific answer-seeking-task, there exists a more general context. In the feasibility study, we might describe such typical user situations and offer specific entry points to link these knowledge needs with appropriate answers. These entry points may include a personalized view to the ontological concepts and relations as well as a list of *top ten questions* which should be easily accessible to users.
- As the volume of knowledge increases it can no longer be assumed that users of knowledge know what is available. User-initiated querying and navigating has to be supplemented by unsolicited delivery of relevant knowledge according to a user profile. In modifying his personal profile, a user may taylor the system to better match his knowledge needs. The profiling may be seen as a user specific view to the ontology.
- The user may be supported in customizing his personal user interface, *e.g.* the entry points (c.f. top-ten-questions) might be changed individually.

#### Possible OTK tool support

The OnToKnowledge consortium makes a number of tools available to support the above mentioned vision of querying a knowledge base:

**Context:** The domain ontology, created with OntoEdit (*cf.* (Staab & Maedche, 2000)) delivers the context. Along the backbone taxonomy, simple generalization or specialization tasks may be offered automatically to the user. Therefore, an inferencing mechanism has to work on the ontological structure. So, also more complex contextual relations and rules may be exploited by the system to deliver semantically rich answers to the user queries.

**Visualization:** The visualization module of AIdministrators WebMaster may be used to present the ontology to the user. The user can select concepts by clicking on the nodes and will additionally be supported by a visualization of the relations between concepts. This graphical user interface, which will be developed in the progress of the OnToKnowledge project from BT Labs, is based on WebMaster.

**Entry points:** The support with appropriate entry points is in close relationship to the navigational paradigms, discussed in the next section.

**User profile:** The creation of user profiles is discussed in Section 3.4.3 about communities of knowledge sharing. The user interface tool based on BT tools in collaboration with the inference system will allow the provision of additional relevant knowledge according to a user profile.

### 3.4.2 Navigate and browse the knowledge base

The fundamental basis of the WorldWideWeb is to provide navigation via hyperlinks. While conceptually simple, finding the right approach is hard to maintain due to different views of users and providers as indicated in the usage scenario in Section 2.2.3

### Vision

Visualization addresses navigation and speed of retrieval within the KM environment and also support analysis of knowledge content. Visualization ranges from tactical navigation using todays desktop metaphor (*e.g.* to select an application as entry point to knowledge content) to strategic navigation using visual knowledge maps to guide navigation among all knowledge assets without regard to their originating application or location.

To be useful and successful, an intranet based knowledge management environment must organize knowledge and assemble it in a consistent, logical, and systematic manner. It must allow users to get to the knowledge that they need in a painless and fast manner. In this environment, we have to differentiate between (i) the navigation through web pages along hyperlinks, (ii) the navigation via any visualization of the domain like sitemaps and (iii) the guidance of users with fuzzy information needs:

(i) The navigation through web pages along hyperlinks has to be supported with a consistent structure of the intranet, based on the knowledge needs as analyzed in the feasibility study of the knowledge management project. The entry point is an *opening page* with high functionality, not just graphically attractiveness. From this point, the user must get a quick guidance to the content via different navigational paradigms:

- topic-based: structuring the domain in topics and typical problem areas, *e.g.* technologies, regulations, etc.
- process-based: breakdown of the knowledge along the typical work processes and tasks.
- role-based: organisation of the domain knowledge by roles.
- product-based: structuring the knowledge around products and services along the value chain.

On the basis of a detailed analysis of the knowledge needs, the most useful navigational paradigms, which have to be supported by the knowledge management system, have to be implemented.

(ii) A visualization of a domain ontology can effectively guide users through very large knowledge spaces, even if the user has only limited domain knowledge.

(iii) An additional navigation support is a *guided dialogue* which is particularly well-suited to dealing with fuzzy information needs, where users are not aware of all the relevant criteria. The user may navigate by answering a series of questions (e.g. regarding the required product features or problem symptoms). The answers impose constraints on the search process.

### Possible OTK tool support

The OnToKnowledge consortium makes a number of tools available to support the above mentioned vision of navigating and browsing through a knowledge base:

**Structuring the intranet:** The ontology, created with OntoEdit, may deliver the structure for the different navigational paradigms.

**Visualization of the domain:** As described in the former section, the visualization module of WebMaster may be used to present the ontology to the user. The user may zoom into the visualized domain and navigate through hierarchies or along relations from one concept to another. This graphical user interface, which will be developed in the progress of the OnToKnowledge project from BT Labs.

**Guided dialogue:** The analysis of specific answers may be performed with the help of the Corporum Tool (*i.e.* tool from OnToKnowledge partner Cognit). The results of the analysis may be the input for a search process.

### 3.4.3 Community of knowledge sharing

#### Vision

As the volume of knowledge increases it can no longer be assumed that users of knowledge know what is available. User-initiated querying and navigating has to be supplemented by unsolicited delivery of relevant knowledge. This requires identifying and categorizing users interests. A collaborative functionality supports the selection of knowledge by and for the user by matching knowledge content and experts to users specific needs and interests. It includes profiles and filters, identification of experts, and alignments of users into communities of knowledge sharing. To set up such an environment, a target knowledge management system has to support the following topics:

- Profiles preserve user context information in the overall knowledge management application. Today, user context is preserved on the user' s device. With profiles of context preserved in the network application, users can walk up to any computer, identify themselves, see their customized workspace and resume working.
- Stored knowledge has to be enriched with metadata about knowledge source, type, author, date of storage, representation format etc. In drawing inferences on top of this metadata, the system may create additional knowledge (*e.g.* if an employee writes a document about a specific technology, he might be an expert regarding that technology).
- The metadata allows the linking between knowledge sources and people. If a user asks about a specific technology, the system may point him to an expert.
- New material has to be checked against stored profiles and the users have to be notified, *e.g.* via knowledge push services, whether the match with their profile is successful. This delivered knowledge may be evaluated by the user what may serve as an input to improve the stored profile.
- Users with similar user profiles may be connected. This enables the creation of communities of knowledge sharing.

#### Possible OTK tool support

The OnToKnowledge consortium makes a number of tools available to support the above mentioned vision of a community of knowledge sharing:

**User profiles:** Based on the ontology, created with OntoEdit, an environment to set up, maintain and evolve user profiles will be built by the OnToKnowledge partner BT.

**Enrich with metadata:** The Corporum tool analyzes documents and annotate them with metadata.

**Drawing inferences:** Based on the OIL language (*cf.* (Fensel et al., 2000)), an inferencing system will use relations and rules of the ontology to generate additional knowledge.

**Push services:** Based on the Knowledge Sharing facility, developed by BT, proactive push services supplement the user-initiated querying.

## 3.5 Supporting use cases

### 3.5.1 Ontology development

Ontologies are core elements of OTK projects. Recognizing this importance we expanded the ontology development to a separate and more detailed section (*cf.* Section 4). There we give an overview about existing ontology development methodologies and sketch the lessons learned from those approaches to design the OTK ontology development process.

### 3.5.2 Annotation of documents

#### Vision

In the user-driven use cases, we described how a knowledge worker gets support in seeking knowledge. The main sources of that knowledge stems from intranet pages, documents in electronic formats and databases. All that sources have to be combined in one XML knowledge base, as we assumed earlier. To fill that knowledge base, we need several different process steps as discussed in Section 3.5.3. The semantic interlinkage between intranets, electronic documents and the knowledge base are annotations. The web pages and documents have to be annotated to explicitely represent the semantics of their contents. In the following, we differentiate three ways for creating annotations:

- Manual semantic annotations: In specific applications, it makes sense to support a knowledge provider with tools which enable manual semantic annotations. Hence, several problems appear, which are in detail described in (Erdmann et al., 2000). We do not consider this approach as useful for the OnToKnowledge methodology.
- Template-based annotation:
  - In typical office documents, like Microsoft Word or Excel, a bunch of metadata is already included. The target knowledge management system has to take advantage of this knowledge.

  - Web-based forms with predefined annotations may serve in specific applications as input devices.

- Semi-automatic annotation: With the help of an information extraction-based appraoch, lexical resources should be directly mapped onto concepts and relations contained in the ontology. A knowledge provider transfers his documents to an information extraction system which extracts the annotation-relevant information. These recognized concepts and dependency relations between concepts may be highlighted as suggested annotations and confirmed or rejected from the knowledge provider.

#### Possible OTK tool support

The OnToKnowledge consortium makes a number of tools available to support the above mentioned vision of the annotation of documents:

**Template generation:** Templates are useful for our purposes, especially in specific scenarios, like skill management etc., where web-based templates may easily be processed by employees. The decision about the use of such templates in specific application tasks must be an explicit result of the feasibility study. The design of the templates is one of the graphical user interface tasks.

**Semi-automatic annotation:** We conceive an information extraction-based appraoch for semi-automatic annotation, which should be implemented on top of the Corporum Tool. Incoming documents may be processed using the information extraction system based on Corporum. Corporum associates single words or complex expressions with a concept from the ontology, connected through the domain lexicon linkage. Recognized concepts and dependency relations between concepts are highlighted as suggested annotations. This mechanism has the advantage that all relevant information in the document with regard to the ontology is recognized and proposed to the knowledge provider, who submits or rejects the proposed annotations.

We still have to detail the technical issues about storing annotations and processing them. One of these questions is, whether annotations are kept separately from documents or whether they are contained within documents. Another question is, if annotation is done by a batch job or immediately while creating documents. In the progress of the OnToKnowledge project, we will give full particulars about these topics.

### 3.5.3 Fill the knowledge base

#### Vision

An initial set up of the system includes the transformation of database contents into the XML knowledge base. We do not discuss this initial setting but focus on the ongoing filling of the knowledge base by the knowledge worker.

**Template based filling** In common office software tools, knowledge worker already use templates to generate letters and fax documents with specific corporate identity design and predefined input areas. These templates and supplementary web based forms may be semantically annotated as described in the previous section. Thus, we fill the knowledge base with semantically annotated knowledge.

**Document upload** The knowledge worker occasionally wants to upload documents to share this knowledge with other users. Our target system needs a functionality to automatically gather the relevant content from the document and annotate it automatically.

#### Possible OTK tool support

As already indicated in the previous section about annotation, we prefer templates, which are useful for our purposes, especially in specific scenarios, like skill management etc., where web-based templates may easily be processed by employees. The design of the templates is one of the graphical user interface tasks. We have to detail the issues about document upload in the progress of the project. A possible solution to process this functionality might be the use of the Corporum tool as an annotation facility during the upload of documents.

### 3.5.4 Maintenance

#### Vision

The maintenance of the OnToKnowledge KM system is primarily an organizational task. There have to be clear regulations about the responsibilities of the stakeholders. The management has to assign the people who manage the different maintenance processes. A knowledge engineer has to gather feedback and experiences to improve the ontology on a regular basis.

Evolving ontologies require intelligent solutions for maintaining ontologies, their dependent annotations and the structure of the knowledge base. In any realistic application scenario, incoming information that is to be annotated does not only require some more annotating, but also continuous adaptation to new semantic terminology and relationships (*cf.* (Staab & Maedche, 2000)). Terms evolve in their meanings, or take on new meanings as new technologies are developed, and as existing ones evolve. This results in problems, (i) if the meaning of ontological elements changes, (ii) if the elements in the ontology become unnecessary and are eliminated, or (iii) if new elements are added to the ontology. Our experiences have shown that annotation and ontology development and maintenance must be considered as a cyclic process. Thus, in a realistic maintenance scenario a feedback loop and tight integration is required, so that new conceptual structures can be added to the ontology for supporting the actual task of annotating knowledge and structuring the knowledge base.

#### Possible OTK tool support

The maintenance of ontologies is supported by OntoEdit. There is additional need for a tool that manages the versioning of ontologies and the interlinking between annotations and evolving ontologies. We will give more details on these requirements in the progress of the project.

# Chapter 4

# Development of Ontology

Ontologies aim at capturing domain knowledge in a generic way and provide a commonly agreed understanding of a domain, which may be reused and shared across applications and groups (*cf.* (**?**)). Ontologies are organized in taxonomies and typically contain modeling primitives such as concepts, relations and axioms. Methodologies that guide the building process of ontologies are now proposed by a few research groups. Due to the fact that ontology engineering is still a relatively immature discipline, each research group employs its own methodology. In the following we will present an overview about methodologies (Section 4.1) and the lessons learned from those approaches to create our specific OnToKnowledge ontology development process in Section 4.2.

## 4.1 Overview about existing Methodologies

In the past years, a few research groups proposed methodology approaches guiding the ontology development process. Uscholds Skeletal Methodology were the first methodological outlines proposed in 1995 on the basis of the experience gathered in developing the Enterprise Ontology (*cf.* (Uschold & King, 1995)). On the basis of the Toronto Virtual Enterprise (TOVE) project, Grueninger and Uschold described ontology development steps in (Uschold & Grueninger, 1996). A method to build an ontology in the domain of electrical networks was presented from (Bernaras et al., 1996) as a part of the Esprit KACTUS project. At the same time appeared METHONTOLOGY (Gomez-Perez, 1996), extended in later papers.

In parallel the philosophical discipline of ontology is evolving towards an engineering discipline. (Guarino & Welty, 2000b) demonstrate how some methodology efforts founded on analytic notions that have been drawn from philosophy can be used as formal tools of ontological analysis.

In the following, we will give a brief overview about these methodologies. In addition, we will sketch some experiences and guidelines made in large practical ontology and thesaurus development projects and conclude in Section 4.2 by summarizing lessons learned in the OnToKnowledge Ontology Development Process. We do not discuss (Grosso et al., 1999), (Swartout et al., 1997) and (Lenat & Guha, 1990) which describe further approaches of ontology development methodologies and tools.

### 4.1.1 Skeletal Methodology

This methodology is based on the experience of building the Enterprise Ontology (*cf.* (Uschold & King, 1995)), which includes a set of ontologies for enterprise modeling. The following guidelines for developing ontologies are proposed:

1. *Identify purpose.* Clarify goal and intended usage of the ontology.
2. *Building the ontology*, which is broken down into three steps:
   (a) *Ontology capture.* Identify key concepts and relationships in the domain of interest. Create precise unambiguous text definitions for such concepts and relationships and identify terms to refer to them. Use a middle-out approach to perform this step, so identify the most important concepts which will then be used to obtain the remainder of the hierarchy by generalization and specialization.
   (b) *Coding.* Represent the knowledge aquired in 2(a) in a formal language.
   (c) *Integrate existing ontologies.*
3. *Evaluation.* Make a judgement of the ontologies with respect to a frame of reference which may be requirements specifications or competency questions.
4. *Documentation.* Document ontologies according to the type and purpose.

This methodology 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, involving the use of brainstorming techniques, is given(*cf.* (López, 1999)). A life cycle is not recommended. There is no guideline about the maintenance of evolving ontologies.

### 4.1.2 KACTUS

The approach of (Bernaras et al., 1996) is developed within the Esprit KACTUS project. One of the objectives of this project is to investigate the feasibility of knowledge reuse in complex technical systems and the role of ontologies to support it. The methodology recommends an application driven development of ontologies. So, every time an application is built, the ontology that represents the knowledge required for the application is built. Three steps have to be taken every time an ontology-based application is built:

1. Specification of the application. Provide an application context and a view of the components that the application tries to model.
2. Preliminary design. Based on relevant top-level ontological categories create a first draft where the list of terms and application specific tasks developed during the

previous phase is used as input for obtaining several views of the global model in accordance with the top-level ontological categories determined. Search for existing ontologies which may be refined and extended for use in the new application.

3. Ontology refinement and structuring. Structure and refine the model in order to arrive at a definitive design.

The methodology offers very little detail and does not recommend particular techniques to support the development steps. Also, documentation, evaluation and maintenance processes are missing (López, 1999).

### 4.1.3 METHONTOLOGY

The METHONTOLOGY framework from (Gomez-Perez, 1996) includes:

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.

The methodology offers detailed support in development-oriented activities except formalization and maintenance and describes project management activities. We will use the METHONTOLOGY framework as a skeleton and taylor it for the specific OnToKnowledge need.

### 4.1.4 Formal Tools of Ontological Analysis

In (Guarino & Welty, 2000b) the authors discuss their approach towards developing a methodology for ontology-based model engineering. The approach is based on four fundamental ontological notions that have been drawn from philosophy: identity, unity, rigidity, and dependence. These meta-properties deliver constraints about the way subsumption is used to model a domain. These constraints clarify misconceptions about taxonomies and give support to bring substantial order to ontologies. We will take into account this approach in the refinement phase of our OnToKnowledge Methodology.

### 4.1.5 Experiences from large practical ontology and thesaurus development projects

In the following, we describe lessons learned during the development of large thesaurus and ontologies. We sketch the results of interviews with the research groups who developed the GermaNet (Kunze & Wagner, 1999) and the GETESS ontologies (Staab et al., 1999) and show some results of a literature study of the development process of the CIA world fact book (Frank et al., 1999).

### GermaNet development University of Tuebingen

GermaNet is a lexical-semantic net that has been developed within the LSD Project at the Division of Computational Linguistics of the Linguistics Department at the University of Tuebingen (Kunze & Wagner, 1999). Currently it is being integrated into the EuroWordNet (EWN), a multilingual lexical-semantic database.

GermaNet relates German nouns, verbs, and adjectives semantically by grouping words belonging to the same concept and by defining semantic relations between concepts. It has much in common with the English WordNet and might be viewed as an on-line thesaurus defining an explicit ontology.

*How they set up the project:* The GermaNet was developed manually according to the Princeton WordNet. The developer team adopted the structure of lexicographical files from WordNet with top level concepts and filled some thousand concepts to this structure in a basically heuristic manner. To close gaps, they compare the text corpora and the domain lexicon. During the development, the typical controversial semantic discussions (*e.g.* about concepts which might be in a superordinate or synonymic relation) occured.

*Some lessons learned:*

- To support the group development, the classification in a top level structure helps to portion out areas for each developer.
- As an approach to solve the controversial semantic discussions, they developed a tool which supports the semantic tagging and evaluates the semantic resource.

### GETESS Project University of Karlsruhe and Partners

GETESS is a system that uses semantic methods and natural language processing capabilities in order to provide comprehensive and easy-to-use access to tourist information in the WWW (Staab et al., 1999). The backbone of the system is a tourist domain ontology.

*How they set up the project:* The three members of the ontology engineering team analyzed several top-level-ontologies and choose the HPKB as upper level structure (*cf.* (**?**)). This structure had to be adjusted in intensive joint workshops. Domain relevant concepts were gathered in brainstorming sessions. The categorization effort was done from one member as a discussion base for the whole group. Then attributes and relations were built into the taxonomic hierarchy. The functionality of the target application was the guideline for that process. Additional concepts were analyzed with the help of information extraction tools and a powerful domain lexicon of the partners.

*Some lessons learned:*

- If the ontology engineer knows exactly the application which has to be supported from the ontology, it is easy to model a baseline taxonomy of concepts.

- Each ontology engineer used another nomenclature (*e.g. has-part* or *HAS PART* or *hasPart*): set templates and describe the common notation.
- The overall picture will get lost in a domain of more than 500 concepts with many relations: use visualization tools.
- In a group development, strict rules have to be established about update procedures and versioning of the ontology.
- Especially in early stages of ontology development, the refinement of the ontology may be supported by document analysis.
- The use of a domain lexicon is recommended to reach a wide coverage of the domain.
- The team would strongly recommend to use development tools with workgroup and versioning functionality.

### CIA World Fact Book Stanford University

The World fact book (WFB) knowledge base has been an experiment in the construction of a large-scale knowledge base from a source authored using semantic markup. The content of the knowledge base is, in large part, derived from the CIA World Fact Book, and covers a broad range of information about the worlds nations. The WFB is a highly structured document with a complex underlying ontology. In the following we cite a description about the ontology modelling process from (Frank et al., 1999).

*How they set up the ontology development:*

**Knowledge Refinement** The knowledge extraction process produces terms, organized into otherwise unstructured groups. For example, the group of industries includes the terms manufacturing and automobile manufacturing. While there are thousands of terms in the WFB, the number of terms in each group is much more limited. A typical group will include less than 500 terms. This makes it possible to organize the terms by hand. This is the primary task of knowledge refinement. The knowledge refinement stage, however, involves three subtasks. First, synonyms are eliminated through the use of simple rewrite rules that establish a preferred term in each set of synonyms (e.g.,(preferred-term oil petroleum)). Second, the remaining unique terms are organized into a taxonomy. This allows a system to answer more concrete queries (e.g., "What fossil fuels are there in Saudi Arabia“ as opposed to "What natural resources are there in Saudi Arabia“). It also allows a system to answer more general queries (e.g., "Are there any fossil fuels in Saudi Arabia“ as opposed to "Is there any petroleum in Saudi Arabia“). Finally, the list elements that the parser was unable to decompose are manually split.

**Taxonomy construction** There are several types of objects for which meaningful taxonomies can be built. They are industries, commodities, natural resources, languages, religions and ethnic groups. The information how to taxonomize these objects is not in the WFB; at this stage they relied exclusively on outside knowledge. Many of the classes in the WFB are non-primitive. That is, they are defined by sufficient conditions. The obvious reason for that is that these classes were originally introduced by people, they are not "natural" classes. When the authors of the WFB associated these classes with countries, they based their decisions on some implicit criteria. For example, the reason for the statement that India s industrial sectors include machinery production is probably that there are a significant number of businesses in India that can be described as producing machinery. These rules are sufficient conditions of membership in the class of machinery-production businesses. Therefore, the machinery industrial sector, viewed as a class of businesses, is a non-primitive class. Many of the classes in the WFB can be naturally organized into taxonomies according to multiple attributes. Oil resources, for example, can be classified into offshore and inland deposits, according to the size and significance of deposit, or how fully they have been exploited. One solution is to classify objects along several orthogonal dimensions, introduce classes for each such dimension, and then subclass each WFB class from the appropriate dimensions. For example, the WFB class "small unexploited deposits of iron" is a subclass of iron deposits, of low-abundance natural resources, and of unexploited natural resources. Another way to accomplish this would be to introduce attributes for these dimensions.

**Ontology development goals and criteria** Taking into account that taxonomies of these types of objects can have their own values, they decided to distinguish between two separate tasks: taxonomizing the terms in the WFB, and building a useful ontology of, say, natural resources. The goal was to do both. However, many of the terms in the WFB are not useful enough to be present in a general-purpose ontology. Some of them are direct intersections of other classes ("small deposits of coal"), and some terms are just too bizarre ("two small steel rolling mills" as an industrial sector of Saudi Arabia). To achieve both goals, the result of this stage of knowledge base building was subdivided into two smaller subontologies for all taxonomies constructed. The terms that were present in the WFB, but had no particular value as classes in a general-purpose ontology were exiled to one of the subontologies. Each of these terms is either an alias for a term in the main ontology, or a subclass that was considered too narrow to be retained (e.g. "Coastal-Climate-And-Soils-Allow-For- Important-Tea-And-Citrus-Growth" was abstracted to "fertile soil").

## 4.2 Ontology development process

All the methodologies and practical ontology development experiences have in common that they start from the identification of the purpose of the ontology and the need for domain knowledge acquisition. While the Skeletal Methodology proposes coding in a formal language as a next step, METHONTOLOGY proposes expressing the knowledge as a set of intermediate representations (IR). Then the ontology is generated using translators. We follow this proposal since these intermediate representations (sketched in *competency questionnaires*) provide a user-friendly approach for both knowledge acquisition and evaluation by knowledge engineers and domain experts. The need for evaluation is also identified in all methodologies, but they do not state convincingly how it should be carried out.

In our approach of the ontology development process we will integrate the lessons learned of practical experiences into the steps to be taken to perform each ontology development activity, the techniques used, the output products and how they are to be evaluated. The path of an OnToKnowledge ontology development process is sketched in

Figure 4.1. The feasibility study supports on the one hand the go / no go decision for a concrete knowledge management project and supports on the other hand the first draft of a baseline taxonomy in the ontology development kickoff phase. In the refinement stage, the baseline taxonomy will grow and develop into an ontology with the support of OnTo-Knowledge Tools. The ontology evaluation and maintenance processes covers the issues of evolving ontologies.

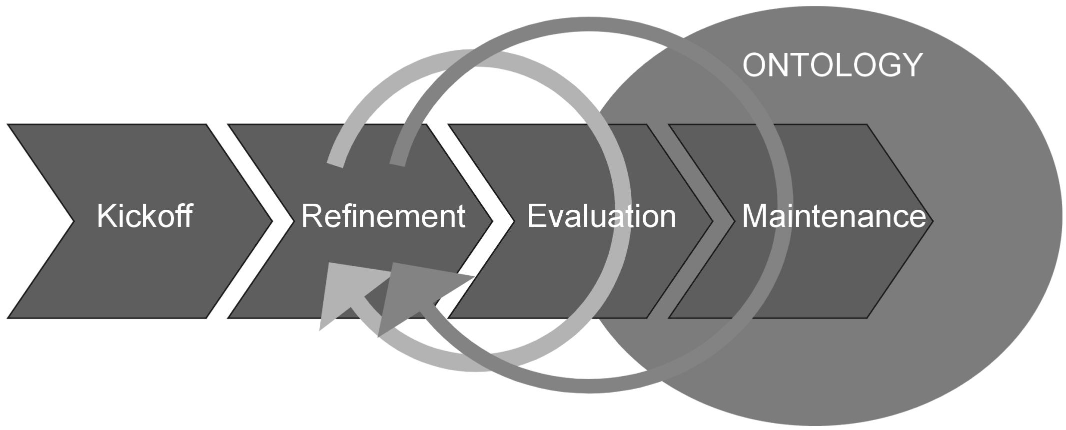


Figure 4.1: The Ontology Development process

### 4.2.1 Kickoff phase

The output product of the kickoff phase is the ontology requirements specification document. The tasks of this phase have to be done by an ontology engineer, who also worked on the feasibility study due to the fact that several outcomes of the feasibility study are a direct input into this phase. The requirements specification document describes what an ontology should support and sketches the planned area of the ontology application. It should also guide an ontology engineer to decide about inclusion, exclusion and the hierarchical structure of concepts in the ontology. It contains the following information (see also an example of a ontology requirements specification document in Figure 4.2):

1. **Goal of the ontology** The feasibility study made clear proposals about interesting areas to be supported by a knowledge management project. The ontology engineer may use the outcomes of the task analysis to describe the goal of the ontology. The following list gives some examples: "The ontology serves as a means to structure the xy domain“, "The ontology serves as a guideline for the knowledge distribution between department A and department B“, "Ontology serves as a base for semantic search“.

2. **Domain and scope** Short description of the domain in use. The description also contains an estimation of the number of concepts and the level of granularity of the planned model. This estimation is based on the knowledge item analysis, a further outcome of the feasibility study.

Ontology Requirements Specification Document

| | |
|---|---|
| **Domain**: | Skill Management and Human Ressource Development at Company ABC |
| **Date**: | 2000/08/22 |
| **Ontology Engineer**: | T. Model |

**Goal**:
- The ontology serves as a guideline for the knowledge distribution between HR department and project managers
- Ontology serves as a base for semantic search for employee skills

**Domain and Scope**:
The ontology contains concepts in the domain of human ressource development. Skills in technologies and market domains of ABC have to be modelled in detail, the common business administration area has to be modelled on a more general level. Similar domain models contain about 400 – 600 concepts.

**Supported Applications**:
Intranet based Skill Management System

**Knowledge Sources**:
- HR department web pages
- Handbook about employee development, internal document
- Technology and product road map, internal document
- Interviews with HR department and project managers

**Users and Use cases**:

| | |
|---|---|
| G. Peoplefind, Human Ressource Department; | attached use case 1 |
| H. Projectering, Project Manager | attached use case 2 |
| B.Boss, Project Manager | attached use case 3 |

**Competency Questions**:
Attached CQ 1

Figure 4.2: Ontology requirements specification document

3. **Applications supported by the ontology** Draft of the ontology based knowledge management application and its system and software environment. The ontology engineer may here as well use the task analysis from the feasibility study as an input source to describe the proposed system and analyze the role of the ontology.The draft must also deliver a clear picture about the ontology interface to the user and answer the following question: what part of the ontology, namely concepts and relations are visible to the user and how does he use them?

4. **Knowledge sources** The knowledge item analyses from the feasibility study serves as an important knowledge source at hand. The ontology engineer may here derive people and documents to complete the list of knowledge sources for the domain in use. The following shows a partial list of knowledge sources as an example:

   - domain experts
   - internal documents
     - dictionairies
     - index lists
     - regulations
     - standard templates
     - product and project descriptions
     - technology white paper

  - telephone index
  - web pages / site statistics
  - SiteSeer web page analysis
  - organization charts
  - employee role descriptions
  - business plan
- external documents

5. **(Potential) users and usage scenarios** List of potential users or user groups and description of each usage scenario. These scenarios should be described from the potential user who may report from own experiences: In what situation did they wish such a system (better search for information, information distribution etc.)? How did they proceed without it? What were the hindering blocks? How would they like to be supported? The usage scenarios sketch the point of view of each individual user, which may vary in between to an extreme degree. Those views give interesting input to the structure of the ontology and the conceptualization. The description of the hindering blocks are also important hints for the design of the ontology based system. The acquisition of the usage scenarios is done via structured or informal interviews.

6. **Competency questions** The usage scenario describe the real existing domain of the targeted system. They deliver information about concepts and their relations which have to be modeled in the target ontology. To derive that information out of the use cases, the ontology engineer has to transform the scenarios in detailed competency questions (Uschold & Grueninger, 1996). This represents an overview of possible queries to the system, indicating the scope and content of the domain ontology. Figure 4.3 shows an example of a competency questionnaire.

The usage scenario/competency question method follows usually a top-down-approach in modeling the domain. With the support of an automatic document analysis as described in the refinement phase, a typical bottom-up-approach may be applied. Both approaches have advantages and drawbacks. The competency questions lead to a more detailled description of the problem area at hand. This supports the fine tuning of the ontology. On the other hand this gathering of several views is never complete. So, this top-down-approach meets the representation of the "information demand" better than the bottom-up-approach with automatic analysis of documents, what itself supports a better representation of the "information supply".

### 4.2.2 Refinement phase

In the refinement phase a first version of a baseline taxonomy (seed taxonomy) will be developed into a seed ontology and expands during the refinement process to a target ontology. This phase is divided into in four subphases: (1) the gathering of a seed taxonomy, (2) the knowledge elicitation process with domain experts based on the initial input from the seed taxonomy to develop the seed ontology, (3) the conceptualization and

| Competency Questionnaire Nr. A | | | |
|---|---|---|---|
| **Domain**: | Skill Management and Human Ressource Development at Company ABC | | |
| **Date**: | 2000/08/22 | | |
| **Ontology Engineer**: | T. Model | | |
| **CQ Nr.** | **Competency Question** | **Concepts** | **Relations** |
| CQ1 | Which of our consultants has experience with JAVA programming language? | Consultant *is a* employee, JAVA *is a* programming language, programming language *is a* experience, experience *is a* skill | Consultant *has* experience |
| CQ2 | Which of our programmers worked in a project with a customer from the chemical industry? | Programmer *is a* employee, Project, Customer, Chemical industry *is a* industry, industry knowhow *is a* experience | Programmer *works in* project, project *at a* customer, Programmer *has* experience |
| CQ3 | Is there a document about our employee accident insurance system? | Document *is a* knowledge source, employee accident insurance(eai) *is a* accident insurance (ai) *is a* insurance, eai system *is a* ai system *is a* insurance system *is a* system | Document *contains* knowledge, employee *has* insurance, company *has* employee accident insurance system |
| CQ4 | What is the average salary for senior programmers? | Senior programmer *is a* programmer *is a* employee, average salary *is a* salary | Employee *has* salary |

Figure 4.3: Competency questionnaire

formalization phase to transfer the seed ontology into the OIL (*cf.* (Fensel et al., 2000)) representation with concepts, attributes, relations and axioms, (4) the refinement phase to use additional tools to improve the target ontology.

1. **Gather a seed taxonomy** One of the major roles of taxonomies is to manifest the backbone structure of an ontology. (Guarino & Welty, 2000a) have found that a "natural result of our analysis is the identification of special properties (nodes) in a taxonomy that best fill this role. These properties form what we call the backbone taxonomy“. In the following, we present some steps to gather, structure and clean such a backbone of what we call seed taxonomy. In this first step of the refinement phase, the ontology engineer may use the ontology editor *OntoEdit* (Staab & Maedche, 2000) as a supporting tool to design a seed taxonomy. The competency questionnaires as a result of the kickoff phase are the basic input documents to gather an initial list of concepts and set the identified is-a relations. The ontology engineer may add missing concepts in hierarchical relations and include obvious generalizations or specifications of concepts.

   This seed taxonomy is now used as input for an automatic document analysis. The OnToKnowledge consortium will develop a tool based on the Corporum system, supporting the extraction of concepts and basic relations out of documents relevant for the domain. In a next step, the resulting list of concepts is integrated into the domain lexicon of the ontology editor. The ontology engineer now adds with simple click-and-select actions additional concepts into the seed taxonomy.

A common problem of ontologies is that their taxonomic structure if often poor and confusing what is typically exemplified by the unrestrained use of subsumption to accomplish a variety of reasoning and representation tasks. In (Guarino & Welty, 2000b) a formal ontology of unary properties is proposed to help using the subsumption relation in a disciplined way. This formal ontology is based on four fundamental philosphical notions (cf. identity, unity, rigidity, and dependence) which impose constraints about the way subsumption is used to model a domain. These constraints clarify misconceptions about taxonomies and give support to bring substantial order to ontologies. The authors present a six step methodology to produce well-founded taxonomies. These steps are important to get a cleaner taxonomy due to the semantic constraints imposed on the is-a relation. In addition, the rigorous analysis forces the ontology engineer to make ontological commitments explicit, clarifying the intended meaning of the concepts used and producing therefore a more reusable taxonomic structure.

2. **Develop the seed ontology** To expand the taxonomy into an ontology, we have to add different types of relations to the taxonomic hierarchy. This knowledge about the domain is captured from domain experts in a knowledge elicitation process based on the initial input from the seed taxonomy. In the following, a way to instantiate the seed ontology is proposed.

   The ontology engineer adds the different types of relations as analyzed in the competency questions to the taxonomic hierarchy in *OntoEdit*. With this basic ontology as a discussion foundation, the ontology engineer starts interviewing the domain experts. They get a visualization of the taxonomic hierarchy with the task to add attributes to concepts and to draw relations between concepts. The ontology engineer documents the additions and remarks in the ontology editor *OntoEdit*.

   Guideline: Make intensive use of the documentation feature of *OntoEdit*

3. **Conceptualization and Formalization** To transfer the seed ontology into the formal OIL representation with concepts, attributes and relations, the ontology engineer may simply use the OIL export function of *OntoEdit*.

4. **Tool supported refinement** In various iterations, the ontology engineer has to refine the ontology to get a target ontology for the application. 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 results as input for the refinement phase.

   In the former phases of the ontology refinement, the use of *OntoEdit* accelerates and simplifies ontology construction. However, the wide-spread usage of ontologies is still hindered by the time-consuming and expensive manual construction task. Within On-To-Knowledge our work evaluates semi-automatic ontology construction from texts as a supplementary approach to ontology engineering. 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 seems to be promising. Therefore a number of proposals have been made to facilitate ontological engineering through automatic discovery from domain data, domain-specific natural language texts in particular (*cf.* (Maedche & Staab, 2000)). The extraction of ontologies from text can have additional benefits for On-To-Knowledge as the required semantic annotation of documents could be provided as a side effect of the extraction process.

The approach is based on different heterogeneous sources: First, the seed ontology may be used as a top level structure for the domain-specific target ontology. Second, domain-specific concepts are acquired using a dictionary that contains important corporate terms described in natural language. Third, a domain-specific and a general corpus of texts may be used to remove concepts that were domain-unspecific. This task is accomplished using the heuristic that domain-specific concepts are more frequent in the domain-specific corpus.

The OnToKnowledge consortium will develop such a tool environment as an interaction between *OntoEdit* and the *Corporum* Tool. A graphical support tool, indicating relations between documents may be of further help for the ontology engineer. This can be attained by an interaction of the proposed toolset with the visualizing components from AIdministrators *WebMaster* System.

Based on the use of those semi-automatic ontology construction tools, gaps in the ontology may be filled and a wider coverage of the domain can be achieved. This refined ontology has to be evaluated in a next step.

### 4.2.3 Evaluation phase

To describe the evaluation task, we cite (Gomez-Perez, 1996): "to make a technical judgement of the ontologies, their associated software environment, and documentation with respect to a frame of reference... The frame of reference may be requirements specifications, competency questions, and/or the real world."

In a first step, the ontology engineer checks, whether the target ontology suffices the ontology requirements specification document (*c.f.* Section 4.2.1)and whether the ontology supports or "answers" the competency questions, analyzed in the kickoff 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 for refinement may be as well the usage patterns of the ontology. The 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.

The evaluation phase will be investigated in more detail in the further process of the case studies of the OnToKnowledge project. A supporting tool for this evaluation phase may be the visualization of usage patterns of the ontology in direct match to the total target ontology. Such a tool may be developed on base of Aidministrators *WebMaster* system.

### 4.2.4 Maintenance phase

The maintenance of ontologies is in close relation to the other maintenance tasks described in Section 3.5.4. We stretch that this task is primarily an organizational process. There have to be strict rules to the update process of ontologies. 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 all possible effects to the application, especially the support use cases *annotation* and *fill knowledge base*.

The maintenance phase also will be investigated in more detail in the further process of the OnToKnowledge project. A supporting tool for this phase may be an ontology server, which supports versioning and group development issues.

# Chapter 5

# Summary

## 5.1 From usage scenarios to use cases

In a typical knowledge management environment, the end user *viz.* a *knowledge worker* has to *Seek knowledge* to perform her knowledge-intensive tasks. At this she encounters usage scenarios like the ones we described in Section 2: knowledge sharing, navigating and querying for knowledge as shown in Figure 5.1.

Communities of knowledge sharing usually consist of a bunch of pull services *i.e.* users actively seek information, they actually need to go and get what they need to know. In our analysis we found an additional use case, *viz. Push services* to provide knowledge to knowledge workers. Those knowledge worker usage scenarios were discussed in Section 3.4 about user-driven use cases.

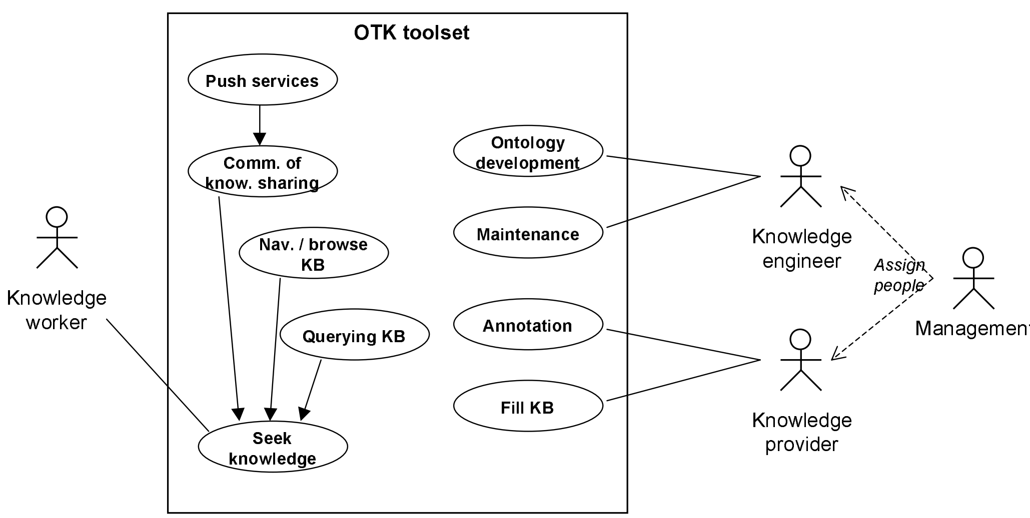


Figure 5.1: From usage scenarios to processes — use cases

For finding knowledge, it first has to be provided, structured and maintained. Several more stakeholders are involved in those service providing functions: *Knowledge engineer*, *Knowledge provider* and *Management* as shown in Figure 5.1, called the *supporters of the system*. The management has several important roles in a KM project: it supports the legitimacy of the project and brings in vision that correlates with the overall company-wide vision. The management needs to be thoroughly convinced of the worth of the project (Tiwana, 2000). Therefore it commits the resources needed — and especially assigns people needed for the supporting use cases.

To provide knowledge efficiently and effectively, the OTK toolset helps knowledge providers to present knowledge *viz.* to *Fill Knowledge Base* and subsequently to enrich the filled-in knowledge with *Annotations*. Those usage scenarios of the supporting stakeholders were discussed in Section 3.5 about supporting use cases.

A well-known means to structure knowledge domains are ontologies, which aim at capturing domain knowledge in a generic way and provide a commonly agreed understanding of a domain, which may be reused and shared across applications and groups. Due to the fact that ontologies are a core element of the OTK toolset, the development process of ontologies were highlighted separately in Section 4.

## 5.2 OTK tool support for user-driven processes

Figure 5.2 shows possible OnToKnowledge tool support for the primary processes (*cf.* Section 3.4).

| | Querying the KB | Navigate and browse KB | Community of knowledge sharing | Push services provided to users |
|---|---|---|---|---|
| Inferencing Tool | X | | X | |
| Visualization Tool | | X | | |
| QL Engine | X | | | |
| User Interface | X | X | X | |
| Knowledge Sharing Facility | X | X | X | X |

Figure 5.2: OTK tool support for user-driven processes

## 5.3 OTK tool support for support processes

Fugure 5.3 shows possible OnToKnowledge tool support for the secondary processes (*cf.* Section 3.5).

| | Ontology development | Annotation | Maintenance | Fill KB |
|---|---|---|---|---|
| OntoEdit | X | | X | |
| Inferencing Tool | X | | X | |
| Corporum | X | X | X | X |
| Visualization Tool | X | | | |
| Knowledge Sharing Facility | | X | | X |

Figure 5.3: OTK tool support for support processes

# Chapter 6

# Outlook

The baseline version will be employed and evaluated in the set of case studies at the OnToKnowledge Partners. With feedback from the partners and guidance from us in implementing the steps, we will incrementally improve the methodology.

For the user-driven and the supporting use cases we will present more detailed conceptions as guidelines for possible results of the OnToKnowledge project and finally we will provide recommendations on how to support these visions with existing and planned OnToKnowledge tools. These recommendations should be actively discussed in the project, especially within the case studies — and they may change over the time. The next version of our methodology will reflect the changes monitored during progress of the project.

The ontology development methodology will be smoother integrated into the feasibility study and the other supporting use cases. We will give guidelines for the further ontology development toolset.

# Bibliography


Bernaras, A., Laresgoiti, I., & Corera, J. (1996). Building and reusing ontologies for electrical network applications. In *Proceedings of the European Conference on Artificial Intelligence ECAI-96*.

Booch, G., Rumbaugh, J., & Jacobson, I. (1998). *The Unified Modelling Language User Guide*. Addison-Wesley.

Davenport, T. & Prusak, L. (1998). *Working Knowledge - how organisations manage what they know*. Havard Business School Press, Boston, Massachusetts.

Decker, S., Erdmann, M., Fensel, D., & Studer, R. (1999). Ontobroker: Ontology based access to distributed and semi-structured information. In R. Meersman et al. (Ed.), *Database Semantics: Semantic Issues in Multimedia Systems*. Kluwer Academic.

Domingue, J. & Motta, E. (2000). Planet-Onto: From news publishing to integrated knowledge management support. *IEEE Intelligent Systems*, 15(3).

Erdmann, M., Maedche, A., Schnurr, H.-P., & Staab, S. (2000). From manual to semi-automatic semantic annotation: About ontology-based text annotation tools. In *COLING-2000 Workshop on Semantic Annotation and Intelligent Content*, Centre Universitaire, Luxembourg.

Fensel, D. (2000). *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer Verlag, Berlin.

Fensel, D., Van Harmelen, F., Decker, S., Erdmann, M., & Klein, M. (2000). OIL in a nutshell. In Dieng, R. (Ed.), *Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference (EKAW-2000)*, Lecture Notes in Artificial Intelligence, LNAI. Springer-Verlag.

Frank, G., Farquhar, A., & Fikes, R. (1999). Building a large knowledge base from a structured source: The CIA world fact book. *IEEE Intelligent Systems*, 14(1).

Gomez-Perez, A. (1996). A framework to verify knowledge sharing technology. *Expert Systems with Application*, 11(4):519–529.

Gomez-Perez, A. & Benjamins, R. (1999). Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods. In *Proceedings of the IJCAI-99 workshop on ontologies and PSM*, Stockholm, Sweden.

Grosso, E., Eriksson, H., Fergerson, R. W., Tu, S. W., & Musen, M. M. (1999). Knowledge modeling at the millennium: the design and evolution of PROTEGE-2000. In *Proceedings of the 12th International Workshop on Knowledge Acquisition, Modeling and Mangement (KAW-99)*, Banff, Canada.

Guarino, N., Masolo, C., & Vetere, G. (1999). OntoSeek: Content-based access to the web. *IEEE Intelligent Systems*, 14(3).

Guarino, N. & Welty, C. (2000a). A formal ontology of properties. Technical report, LADSEB/CNR Technical Report 01/2000. available from http://www.ladseb.pd.cnr.it/infor/ontology/Papers/OntologyPapers.html.

Guarino, N. & Welty, C. (2000b). Identity, unity, and individuality: Towards a formal toolkit for ontological analysis. *Proceedings of ECAI-2000.* available from http://www.ladseb.pd.cnr.it/infor/ontology/Papers/OntologyPapers.html.

Kunze, C. & Wagner, A. (1999). Integrating germanet into eurowordnet, a multilingual lexical-semantic database. *Sprache und Datenverarbeitung - International Journal for Language Data Processing*, 23(2).

Lenat, D. & Guha, R. (1990). *Building large knowledge-based systems. Representation and inference in the CYC project.* Addison-Wesley, Reading, Massachusetts.

López, F. (1999). Overview of methodologies for building ontologies. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends.* CEUR Publications.

Luke, S., Spector, L., Rager, D., & Hendler, J. (1997). Ontology based web agents. In *Proceedings of the First International Conference on Autonomous Agents.*

Maedche, A. & Staab, S. (2000). Discovering conceptual relations from text. In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000)*, Berlin. IOS Press.

Perkowitz, M. & Etzioni, O. (1997). Adaptive web sites: an AI challenge. In *Proceedings of the 15th International Joint Conference on AI (IJCAI-97)*, Nagoya, Japan.

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

Staab, S., Braun, C., Bruder, I., Duesterhoeft, A., Heuer, A., Klettke, M., Neumann, G., Prager, B., Pretzel, J., Schnurr, H.-P., Studer, R., Uszkoreit, H., & Wrenger, B. (1999). A system for facilitating and enhancing web search. In *Engineering Applications of Bio-Inspired Artificial Neural Networks. IWANN '99 - Proceedings of International Working Conference on Artificial and Natural Neural Networks*, LNCS 1607, pages 706–714, Alicante, ES. Springer Verlag.

Staab, S. & Maedche, A. (2000). Ontology engineering beyond the modeling of concepts and relations. *Proceedings of he ECAI'2000 Workshop on Applications of Ontologies and Problem-Solving Methods.*

Swartout, B., Ramesh, P., Knight, K., & Russ, T. (1997). Toward distributed use of largescale ontologies. In *Symposium on Ontological Engineering of AAAI*, Stanford, CA.

Tiwana, A. (2000). *The Knowledge Management Toolkit.* Prentice Hall PTR., Upper Saddle River, NJ, USA.

Uschold, M. & Grueninger, M. (1996). Ontologies: Principles, methods and applications. *Knowledge Sharing and Review*, 11(2).

Uschold, M. & King, M. (1995). Towards a methodology for building ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95*, Montreal, Canada.

van Harmelen, F. & Fensel, D. (1999). Practical knowledge representation for the web. In *Proceedings of the IJCAI-99 Workshop on Intelligent Information Integration.* CEUR Publications.