Final Report

Dieter Fensel, Frank van Harmelen, Ying Ding, Michel Klein, Peter Mika,
Hans Akkermans,
Jeen Broekstra, Arjohn Kampman, Jos van der Meer,
Rudi Studer, York Sure,
John Davies, Alistair Duke,
Robert Engels,
Victor Iosif, Atanas Kiryakov,
Thorsten Lau, Ulrich Reimer, and
Ian Horrocks

Identifier	**Final Project Report**
Class	**Deliverable**
Version	**3.0**
Version date	**November 2002**
Status	**draft**
Distribution	**Public**
Responsible Partner	**Vrije Universiteit Amsterdam**

Ontoknowledge Consortium

This document is part of a research project 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 (VU) (co-ordinator), NL; the 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; and Sirma AI, Ltd. - Artificial Intelligence Labs, OntoText Lab., Bulgaria.

Vrije Universiteit Amsterdam (VU)
Faculty of Sciences, Division of Mathematics and Computer Science
De Boelelaan 1081a
1081 HV Amsterdam, the Netherlands
Fax and Answering machine: +31-(0)20-872 27 22
Contact person: 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
Contact person: 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
Contact person: Ulrich Reimer
E-mail: Ulrich.Reimer@swisslife.ch

British Telecommunications plc
BT Adastral Park
Martlesham Heath
IP5 3RE Ipswich, UK
Tel: (44 1473)640299, Fax: (44 1473)642459
Contact person: 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
Contact person: Bernt. A.Bremdal
E-mail: bernt@cognit.no

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

AIdministrator Nederland BV
Julianaplein 14B
3817CS Amersfoort, NL
Tel: (31-33)4659987, Fax: (31-33)4659987
Contact person: Jos van der Meer
E-mail: Jos.van.der.Meer@aidministrator.nl

Sirma AI, Ltd. – Artificial Intelligence Labs, OntoText Lab.
38A, CHRISTO BOTEV BLVD
Sophia 1000, Bulgaria
Tel: (359 2)981 23 38, Fax: (359 2)981 90 58
Contact person: Atanas Kiryakov
E-mail: naso@sirma.bg

Acknowledgement

We would like to thank our project officer Vincent Obozinski. He always help us to bypass the bureaucratic cliffs in running an IST project. Also he was very interested and supportive in regard to the topic of the project. We also would like to mention our reviewers Wolfram Brandes, Nicola Guarinno, and Robert Meersman. They had to join four review meetings and had to read thousands of pages. In return, they provided as with many valuable ideas and criticisms that help us in improving the performance of the project. Finally, we should not forget to thank Jim Hendler, who managed to organize a similar research project in the U.S.A. with an order of magnitude higher funding. It was the collegial cooperation with his projects that helped us significantly to reach a critical mass at the dissemination site.

1 Overall Achievements and Results

Ontoknowledge provides several breakthroughs on our way to the next generation web: advanced tools, a ontology language standard for the web, case studies, and knowledge management methodology.

1.1 Tools

A key deliverable of the Ontoknowledge project is the resulting software toolset. Several consortium partners are participating in the effort to realize in software the underpinning ideas and theoretical foundations of the project. A major objective of the project is to create intelligent software to support users in both accessing information and in the maintenance, conversion, and acquisition of information sources. These tools are based on a three-layered architecture around information access, information storage, and information generation.

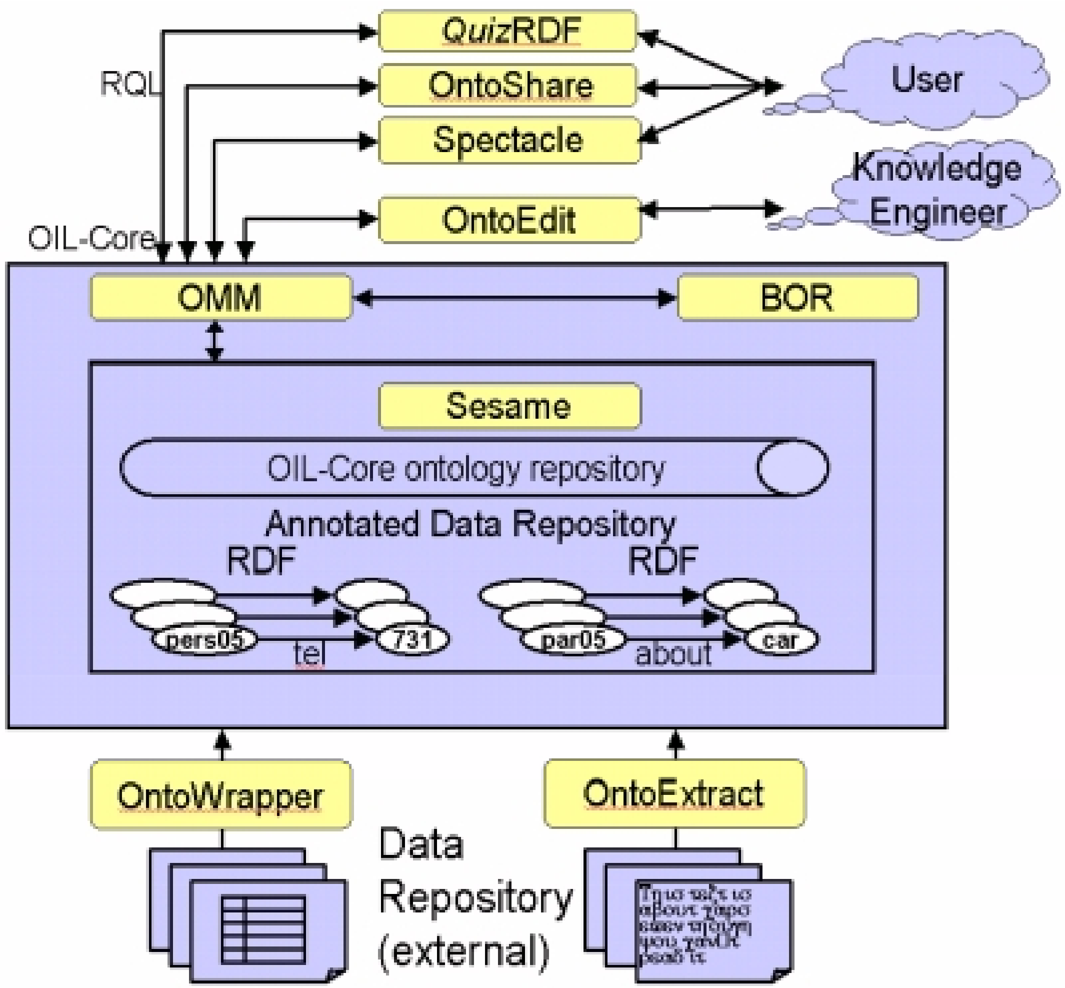

Fig. 1 The technical architecture of Ontoknowledge .

More detailed descriptions of most of the tools can also be found in the appendix.

QuizRDF is an ontology-based tool for knowledge discovery which combines traditional keyword querying of WWW resources with the ability to browse and query against RDF annotations of those resources. RDF(S) and RDF are used to specify and populate an ontology and the resultant RDF annotations are then indexed along with the full text of the annotated resources. The resultant index allows both keyword querying against the full text of the document and against the literal values occurring in the RDF annotations, along with the ability to browse and query the ontology. This ability to combine searching and browsing behaviors more fully supports a typical information-seeking task than "traditional" search engine technology. The approach is characterized as "low threshold, high ceiling" in the sense that where RDF annotations exist they are exploited for an improved information-seeking experience but where they do not yet exist, a search capability is still available.

OntoShare enables the storage of best practice information in an ontology and the automatic dissemination of new best practice information to relevant co-workers. It also allows users to browse or search the ontology in order to find the most relevant information to the problem that they are dealing with at any given time. The ontology helps to orientate new users and acts as a store for key learning and best practices accumulated through experience.

Spectacle organizes the presentation of information. This presentation is ontology driven. Ontological information, such as classes or specific attributes of information, is used to generate exploration contexts for users. An exploration context makes it easier for users to explore a domain. The context is related to certain tasks, such as finding information or buying products. The context consists of three modules:

- Content: specific content needed to perform a task,
- Navigation: suitable navigation disclosing the information,
- Design: applicable design displaying the selected content.

OntoEdit makes it possible to inspect, browse, codify and modify ontologies, and thus serves to support the ontology development and maintenance task. Modelling ontologies using OntoEdit involves modelling at a conceptual level, viz. (i) as independently of a concrete representation language as possible, and (ii) using GUI's representing views on conceptual structures (concepts, concept hierarchy, relations, axioms) rather than codifying conceptual structures in ASCII.

The **Ontology Middleware Module (OMM)** can be seen as the key integration component in the OTK technical solution architecture. It supports well-defined application programming interfaces (OMM API) used for access to knowledge and deals with such matters as: Ontology versioning, including branching; Security – user profiles and groups are used to control the rights for access, modifications, and publishing; Meta-information and ontology lookup – support for meta-properties for whole ontologies, as well as for separate concepts and properties; Access via a number of protocols: HTTP, RMI, and SOAP.

From a functional point of view, OMM supports the two major scenarios of usage of the On-To-Knowledge tools as follows:

- Knowledge engineering – the versioning system makes OMM/Sesame ideal environment for collaborative KE, enabling development style similar to the one which the source control systems (CVS) provide for software development. The Sesame/OMM plug-in for

OntoEdit allows multiple knowledge engineers to use the editor as a front-end, downloading the latest version of the ontology from OMM and uploading their contributions. In order to let this scenario work, OMM silently does smart merging as well as tracking of the changes introduced. At each moment, the updates can be listed and older version can be retrieved;

- Knowledge use – the access control sub-system of OMM makes possible definition of fine grained security policies which can capture fairly complex business logic. This unique feature, combined with the easy integration (because of the multi-protocol support for the API) makes OMM ideal back-end for enterprise knowledge management applications.

- Following the spirit of the On-To-Knowledge toolset, OMM integrates tightly only with the Sesame repository and the OntoEdit editor, but provides guaranteed interoperability with the rest of the tools.

Sesame is a system that allows persistent storage of RDF data and schema information and subsequent online querying of that information. Sesame has been designed with scalability, portability and extensibility in mind. One of the most prominent modules of Sesame is its query engine. This query engine supports an OQL-style query language called RQL. RQL supports querying of both RDF data (e.g. instances) and schema information (e.g. class hierarchies, domains and ranges of properties). RQL also supports path-expressions through RDF graphs, and can combine data and schema information in one query.

BOR provides additional reasoning services so as to extend the functionality provided by SESAME. Most of the classical reasoning tasks for description logics (DL) are available, including realization and retrieval. The goal was to enable even wider set of applications, such as information extraction and automatic ontology integration. A strategy called pre-reasoning was used to implement workarounds for a number of logical problems proven to be computationally intractable for languages as expressive as OIL.

Information extraction and ontology generation is performed by means of the **CORPORUM** toolset (OntoWrapper and OntoExtract) and is situated in the extraction layer. CORPORUM has two related, though different, tasks: interpretation of natural language texts and extraction of specific information from free text. Whereas CORPORUM tools can perform the former process autonomously, the latter task requires a user who defines business rules for extracting information from tables, (phone) directories, home-pages, etc. Although this task is not without its challenges, most effort focuses on the former task, which involves natural language interpretation on a syntactic and lexical level, as well as interpretation of the results of that level (discourse analysis, co-reference and collocation analysis, etc.).

1.2 Baseline technology and beyond

In this section we given an account of what was developed and or integrated in the project on top of the baseline technologies/products (background knowledge).

OntoWrapper: no baseline technology, developed fully under On-To-Knowledge

OntoExtract: baseline technology =

- main engine
- natural language analysis algorithms
- concept extraction technology

Development under On-To-Knowledge:

- more flexible main engine to enable user to influence its behaviour
- export in RDF(S) / DAML+OIL format
- usage of background knowledge during concept extraction
- communication with Sesame

TableAnalyser: no baseline technology, developed fully under On-To-Knowledge (but discontinued after cancellation of Swiss Life case study)

Sesame: no baseline technology, developed fully under On-To-Knowlege

OMM: no baseline technology, developed fully under On-To-Knowlege

BOR: no baseline technology, developed fully under On-To-Knowlege

OntoEdit: baseline technology = single user ontology editing environment;
Development under On-To-Knowledge:

- integration with methodology stages
- *Sesame plugin* for transparent communication with remote repository
- *Mind2Onto* tool for uploading lightweight ontologies developed under MindManager
- *OntoKick plugin* for capturing competency questions
- *OntoFiller plugin* for generating different natural language lexical representations
- *OntoClean pluging* (preliminary) for verifying against OntoClean methodology (work by Guarino et al).

Spectacle: Baseline technology =

- clustermap viewer
- generating structured web-sites from semi-structured external data-sources

Development under On-To-Knowledge:

- development of re-usable Sesame Client library for using data from Sesame repository
- development of special purpose concept-graph viewer for concept structures produced by OntoExtract.
- development of RDF Explorer (generic textual RDF-browser)
- many technical improvements to the core engine to deal with requirements of the case-studies and usage of the data produced by OntoExtract

OntoShare: baseline technology = Jasper knowledge-sharing system and ProSum text summariser;
Development under On-To-Knowledge:

- complete system redesign and reimplementation in Java
- addition of hierarchical interest groups
- addition of RDF-based data model

- addition of capability for ontology to evolve based on usage
- addition of capability to import/export RDF information resources
- Integration of SQL database for back-end data storage

QuizRDF: baseline technology = WebFerret (strictly text-based information retrieval web-engine)
Development under On-To-Knowledge:

- extending WebFerret to be aware of meta-data in RDF format
- using this meta-data for indexing and retrieving documents
- redesigning the user-interface to allow for a interleaved usage of text-based and ontology-based document retrieval

1.3 Scalability

In order to assess the *scalability* of the Ontoknowledge tools, the following orders of magnitude have been provided as requirements in deliverable X.1:

1. In our case-studies, ontologies are expected to be $O(10^3)$ classes. This number is relevant for the ontology-editor and for the ontology-storage.
2. Intranets are expected to be up to $O(10^4)$ pages.
3. QuizRDF can index up to $O(10^6)$ pages.
4. The ontology-extraction service from CognIT analyses O(1) pages per second.
5. The same ontology-extraction service generates O(10) – $O(10^2)$ RDF triples per page, and more likely to be O(10) then $O(10^2)$.
6. Combining [2] and [5] results in $O(10^5)$ triples to store in Sesame (with a maximum of $O(10^6)$). The next version of Sesame is expected to be able to cope with this. The current functional, but not scalable, version has already been tested with up to $O(10^5)$ triples.
7. Combining [2] and [4] results in $O(10^4)$ seconds for the ontology extraction service to analyse a complete intranet. This amounts to a few hours to 1 day for a full analysis. This is acceptable since such a full extraction service can be done off-line, and needs to be done at most once a day (and more likely much less often).

The above calculations lead us to believe that the tools that we developed in the Ontoknowledge project are able to deal with the demands of the three case-studies as included in the project.

In our case-studies, we have exercised our tools to exactly these orders of magnitude, with the exception of the size of the ontologies, which have been of $O(10^2)$ classes instead of $O(10^3)$. Also, the intranets from the three case-studies were of the order of $O(10^3)$ pages. All tools were comfortably able to deal with these size-requirements from the case-studies.

No extensive scalability or endurance testing has been performed in the project (nor were we contracted to do so), but experience with these tools in other contexts lead us to the following:

- the indexing of QuizRDF scales up to $O(10^6)$ pages. This is well beyond what is expected for any reasonable KM scenario (short of a web-wide search engine).

- The processing speed of OntoExtract allows processing of up to 60.000 pages per day, again quite sufficient for any reasonable KM scenario. In practicel the bottleneck is the speed of web-servers and their connections, not the speed of the OntoExtract analysis engine.

- new developments on Sesame (not under On-To-Knowledge) are directed to maximising in-memory caching. Early experiments show that the entire Open Directory Project (800Mb of RDF, currently the largest available RDF resource) can be loaded in memory of a dedicated server (2GHz processor, 1Gb memory) within 15 minutes. Query answering on in-memory storage is instantenous. Again, this size of RDF resource is well beyond what is expected for any reasonable KM scenario.

These results lead us to believe that our tools will scale to any reasonable KM scenario. Scaling our (or any other) toolset to full world-wide web size would be a significant technological challenge, and would make the basis for a good targeted research action under the VIth framework.

1.4 OIL: Inference Layer for the Semantic World Wide Web

The **OIL** language is to designed to combine frame-like modeling primitives with the increased (in some respects) expressive power, formal rigor and automated reasoning services of an expressive description logic. OIL also comes “web enabled” by having both XML and RDFS based serializations (as well as a formally specified “human readable” form, which we will use here). As part of the Semantic Web activity of the W3C, a very simple web-based ontology language had already been defined, namely RDF Schema. This language only provides facilities to define class- and property-names, inclusion axioms for both classes and properties (subclasses and sub properties), and to define domain and range constraints on properties. OIL has been designed to be a superset of the constructions in RDF Schema and the syntax of OIL has been designed such that any valid OIL document is also a valid RDF(S) document when all the elements from the OIL-namespace are ignored. The RDF Schema interpretation of the resulting sub document is guaranteed to be sound (but of course incomplete) with respect to the interpretation of the full OIL document.

For many of the applications, it is unlikely that a single language will be ideally suited for all uses and all users. In order to allow users to choose the expressive power appropriate to their application, and to allow for future extensions, a layered family of OIL languages has been described. The sub-language OIL Core has been defined to be exactly the part of OIL that coincides with RDF(S). This amounts to full RDF(S), without some of RDF’s more dubious constructions: containers and reification. Much of the field of Knowledge Engineering and Knowledge-based systems has traditionally focused on rather rich ontologies, using expressive language to express classes, properties of various types, meta-classes, etc. This tradition also underlaid the original design of OIL as a rather rich Description Logic (in fact: one of the richest logics that is still known to be tractable). During the course of the project, the case-studies shed a different light on the requirements for ontology languages in the knowledge management applications that they were involved in: very lightweight ontologies were constructed, sometimes consisting of not much more than taxonomies of terms, often with no properties relating them. Similarly, the automatic concept-extraction technology that was employed also yielded very light-weight collections of terms, with almost no hierarchical structure and with only very general associations between the concepts (of the kind: "$Concept_1$ isStronglyRelatedTo $Concept_2$").

A common finding of all the case-studies was that these lightweight ontologies went quite a long way towards providing the needs of the applications that were built in these case-studies. They worked very well as the basis for search, personalization and navigation. This finding (the usefulness of lightweight ontologies) has had a direct impact on the design of the W3C Web Ontology Language OWL[1] (which is directly based on OIL): the current draft of this language (June'02) includes a sub language OWL Light. This language roughly consists of RDF Schema plus equality plus 0/1-cardinality, and is particularly suited to express light-weight ontologies of the kind that have been driving our case studies.

This is not to say that full OIL (or: full OWL) is not needed in its own right: when the W3C Working Group discussed the option to standardize only OWL Light, there were strong complaints from communities such as medical science and technical engineering, who both voiced strong needs for a language with at least the expressive power of full OIL/OWL, if not more. Apparently, the requirements and circumstances for these communities are substantially different from ours: our case-studies were all situated in business environments, with domain experts that were unskilled in modelling ontologies, and that were working in an environment were business pressure meant that little time and effort could be devoted to modelling rich ontologies. The layered design that has resulted from the above (OWL Light as a sub language of full OWL) reflects the layered structure that was already present in our proposals for OIL from the very beginning.

1.5 Business Applications in Semantic Information Access

Method and tool development in Ontoknowledge was strongly driven by several case studies. Some of them started in the earliest stages of the project so that the project was geared towards practical demands from the very beginning. The case studies also serve as a means to evaluate the project results. Swiss Life carried out two of the case studies. One of these approached the problem of finding relevant passages in a very large document about the International Accounting Standard (IAS) on the extranet (over 1000 pages). Accountants who need to know certain aspects of the IAS accounting rules use this document. As the IAS standard uses very strict terminology, it is only possible to find relevant text passages when the correct terms are used in the query. Very often, this leads to poor search results. Although the ontology that was built is structurally quite simple, it greatly improves search results. Swiss Life's second case study is a skills management application that uses manually

[1] Meanwhile, OIL has been adopted by a joined EU/US initiative that developed a language called DAML+OIL. In November 2001, the W3C started a Web Ontology Working Group for defining a language. This group is chartered to take DAML+OIL as its starting point and is developing a language called OWL.

constructed ontologies about skills, job functions, and education. These consist of 800 concepts with several attributes, arranged into a hierarchy of specializations. There are also semantic associations between these concepts. The skills management system makes it easy for employees to create a personal home page on the company's intranet that includes information about personal skills, job functions, and education.

According to the insight we have gained at Swiss Life during our participation in the Ontoknowledge project Knowledge Management applications typically need ontology languages with a medium degree of expressiveness, reaching from simple taxonomies to hierarchies of concepts with properties and semantic relationships between concepts, including simple cardinality constraints. The more complex ontologies mainly result from the integration with other application systems. For example, the Skills Management application (SkiM) only uses a simple taxonomy for the available skills. However, to realize the application logic a more complex ontology lies underneath, although the user will never see it (see below). An extension of SkiM to cover functionalities like gap analysis, the integration with career planning and keeping track of attended training courses would require an additional degree of detail in the ontology.

In contrast, the IAS application only employs an ontology with associative links between concepts and has only an implicit, rudimentary concept hierarchy (via substring matching between two noun groups, like “insurance” and “annuity insurance”). The IAS application shows that an ontology-based application can generate considerable benefit even with a very simple ontology.

The case study done by EnerSearch AB focuses on validating the industrial value of the project’s results with respect to the needs of a virtual organization. The goal of the case study is to improve knowledge transfer between EnerSearch's in-house researchers and outside specialists via the existing website. The study also aims to help the partners from shareholding companies to obtain up-to-date information about research and development results.

The BT case study was carried out within BT's Research and Development organization - BTexact Technologies - and was focussed on supporting a virtual community of practice. Communities of practice are an important concept in knowledge management since they are a key tool in facilitating the sharing of knowledge around an organization. A community of BTexact personnel in the fields of conferencing, knowledge management and personalization was chosen. These people are researchers, developers and technical marketing professionals. Such individuals need to share knowledge as part of their job - they are often referred to as knowledge workers. Using the OTK methodology, an ontology was elicited from this community, created in OntoEdit and used in OntoShare to facilitate knowledge sharing. The ontology and associated knowledge can then be stored in the SESAME RDF repository for reuse by other tools (e.g. the visualization tool Spectacle, which visualized OntoSahre repositories as part of the user profiling workpackage). A user evaluation exercise using both quantitative and qualitative measures was carried out.

1.6 Methodology for employing ontology-based Knowledge Management Applications

Finally, a methodology for employing ontology-based tools for knowledge management applications was developed by applying an initial baseline methodology (cf. S.) in the case studies and continuously improving it with the insight gained from experience. This methodology provides guidelines for introducing knowledge management concepts and tools into enterprises, helping knowledge providers and seekers to present knowledge efficiently and effectively. Core contributions of the methodology are: (i) An overview of the Ontoknowledge building blocks and their relationships, (ii) a methodology for introducing and maintaining ontology based knowledge management solutions into enterprises with a focus on Knowledge Processes and Knowledge Meta Processes and, last but not least, (iii) the illustration of process steps by examples and lessons learned derived from applying the OTK tool suite in the OTK case studies according to the methodology.

The OTK methodology focusses on the identification, illustration and instantiation of the Knowledge Process and Knowledge Meta Process. To make it a self-contained document, we included the On-To-Knowledge building blocks, i.e. relevant aspects of knowledge management, ontologies and the overall project setting. Beside others, one of the core contributions of this methodology is the linkage of the tool suite with the case studies by showing when and how to use available tools of the OTK tool suite during the process of developing and running the case studies. In a nutshell we have shown the following items:

- A process oriented methodology for introducing and maintaining ontology based knowledge management systems. Core to the methodology are Knowledge Processes and Knowledge Meta Processes. While Knowledge Meta Processes support the setting up of an ontology based application, Knowledge Processes support its usage.
- Tool support offered by the On-To-Knowledge tool suite for the process steps. E.g. numerous plugins for OntoEdit support different steps of the Knowledge Meta Process.
- Illustration of the process steps instantiations by numerous examples derived from the On-To-Knowledge case studies.

The main lessons learned during setting up and employing the methodology in the case studies are:

- Different processes drive KM projects, but ``Human Issues" might dominate other ones (as already outlined by Davenport).
- Guidelines for domain experts in industrial contexts have to be pragmatic.
- Collaborative ontology engineering requires physical presence and advanced tool support.
- Brainstorming is very helpful for early stages of ontology engineering, especially for domain experts not familiar with modelling.

1.7 Conclusions

The Web and company intranets have boosted the potential for electronic knowledge acquisition and sharing. Given the sheer size of these information resources, there is a strategic need to move up in the data – information – knowledge chain. Ontoknowledge takes a necessary step in this process by providing innovative tools for semantic information processing and thus for much more selective, faster, and meaningful user access. The technology developed has been proven to be useful in a number of case studies. We can improve information access in the large intranets of sizeable organizations. The technology has been used to facilitate electronic knowledge sharing and re-use for customer relationship management and knowledge management in virtual organizations.

More information on Ontoknowledge can be found at J.…] and on the project's website *http://www.ontoknowledge.org/*.

It remains to ask how much Ontoknowledge managed to achieve its objectives, i.e., to ask whether Ontoknowledge has been a success or a failure? We will quote from the project proposal:

> "The goal of the Ontoknowledge project is to support efficient and effective knowledge management. We will focus on acquiring, maintaining, and accessing weakly-structured online information sources:
>
> - *Acquiring*: Text mining and extraction techniques are applied to extract semantic information from textual information (i.e., to acquire information).
> - *Maintaining*: RDF and XML are used for describing syntax and semantics of semi-structured information sources. Tool support enables automatic maintenance and view definitions on this knowledge.
> - *Accessing*: Push services and agent technology support users in accessing the information.
>
> The approach of the project is the use of ontologies for each of these three processes. To achieve these advanced goals, we will develop an ontology-based tool and environment. We will develop a three-layered tool environment to achieve these goals. At the lowest level (the information level), weakly-structured information sources are processed to extract machine-processable meta-information from them. The intermediate level (the representation level) uses this meta-information to provide automatic access, creation, and maintenance of these information sources. The highest level (called access level) uses advanced push ad pull techniques to lower the thresholds for accessing this information. Agent-based techniques as well as state-of the art querying and visualization techniques can fully employ the formal annotations to guide user access of information. At all levels, ontologies are the key asset in achieving the described functionality. The methodology will complement the tool helping to bridge the gap between information needs and information sources."

As it became evident from what we described above, this expectation formulated in the project proposal in 1999 describes precisely what we achieved in 2002. The success of

Ontoknowledge can also be seen be its number of successor IST projects[2] that take essential parts of our vision and work for their further implementation:

- **COG: Corporate Ontology Grid, http://www.cogproject.org/.** Its main objective is to demonstrate the applicability of Semantic Web technologies to industry. Technically it focusses on the technological innovation of automatically translating data between data formats by the way of a semantic mapping to a central Ontology.

- **ESPERONTO: Application Service Provision of Semantic Annotation, Aggregation, Indexing and Routing of Textual, Multimedia, and Multilingual Web Content, http://esperonto.semanticweb.org/.** The aim of Esperonto is to bridge the gap between the actual World Wide Web and the Semantic Web by providing a service to "upgrade" existing content to Semantic Web content.

- **FF-POIROT: Financial Fraud Prevention-Oriented Information Resources using Ontology Technology, http://www.starlab.vub.ac.be/research/projects/default.htm#Poirot**. This project aims at compiling for several languages (Dutch, Italian, French and English) a computationally tractable and sharable knowledge repository for the financial forensics domain. This Ontology, or parts of it, may be commercially exploited as a set of Semantic Web services.

- **GRACE: Grid Search and Categorization Engine**.

- **HtechSight: A knowledge management platform with intelligence and insight capabilities for technology intensive industries, http://banzai.etse.urv.es/~htechsight/.** The h-TechSight project builds on existing achievements in the field of knowledge Management in order to advance technology om to principal directions: (1) Utilize standardized knowledge representation frameworks in order to facilitate interoperability, and (2) contribute to the development of the nest generation of Web- enabled KM platforms.

- **InDiCo: Integrated Digital Conferencing.**

- **MONET: Mathematics on the Net, http://monet.nag.co.uk/cocoon/monet/index.html**. The aim of the MONET project is to demonstrate the applicability of the latest ideas for creating a Semantic Web to the world of mathematical software, using sophisticated algorithms to match the characteristics of a problem to the advertised capabilities of available services and then invoking the chosen services through a standard mechanism

- **MOSES: A modular and Scalable Environment for the Semantic Web**.

- **ONTO-LOGGING: Corporate Ontology Modelling and Management System, http://www.ontologging.com/.** The main result of the project development is a technological and functional platform pilot (toolbox) that could be used for the building of new generation of knowledge management products.

---

[2] Semantic Web Technologies in Europe's IST Programme 1998 - 2002, by Hans-Georg Stork, http://www.ercim.org/publication/Ercim_News/enw51/stork.html.

- **OntoWeb: Ontology-based information exchange for knowledge management and electronic commerce, http://www.ontoweb.org/.** The goal of OntoWeb Network is to bring researcher and industrials together enabling the full power ontologies may have to improve information exchange in areas such as: information retrieval, knowledge management, electronic commerce, and bioinformatics. It will also strengthen the European influence on standardization efforts in areas such as web languages (RDF, XML), upper-layer ontologies, and content standards such as catalogues in electronic commerce.

- **ROCKET - Roadmap to Communicating Knowledge Essential for the Industrial Environment, http://rocket.vub.ac.be/.** Rocket will prepare a strategic roadmap for future developments in organisational learning relevant to the education of engineers and knowledge workers.

- **SCULPTEUR: Semantic and Content-Based Multimedia Exploitation for European Benefit.**
- **SEWASIE: Semantic Webs and Agents in Integrated Economies, http://www.sewasie.org/.** The goal of the SEmantic Webs and AgentS in Integrated Economies (SEWASIE) project is the design and the implementation of an advanced search engine allowing information access via a machine-processable semantics of data in order to provide the basis of structured web-based communication.

- **SPACEMANTIX: Combining Spatial and Semantic Information in Product Data**.

- **SPIRIT: Spatially-Aware Information Retrieval on the Internet, http://www.cs.cf.ac.uk/department/posts/SPIRITSummary.pdf.** The main aim of the project is to create tools and techniques to help people to find information that relates to specified geographical locations

- **SWAD-Europe: W3C Semantic Web Advanced Development for Europe, http://www.w3.org/2001/sw/Europe/**. The SWAD-Europe project aims to support W3C's Semantic Web initiative in Europe, providing targeted research, demonstrations and outreach to ensure Semantic Web technologies move into the mainstream of networked computing.

- **SWAP: Semantic Web and Peer-to-Peer, http://swap.semanticweb.org/**. SWAP combines two highly successful technologies, viz. Semantic Web and Peer-to-peer computing. SWAP will develop the technology that is necessary to allow users their individual views on knowledge AND let them share knowledge effectively.

- **SWWS: Semantic-Web-Enabled Web Services, http://swws.semanticweb.org/.** Web Services promise a new level of service on top of the current web. However, in order to employ their full potential, appropriate description means need to be developed. Therefore, the main objectives of SWWS are to provide a comprehensive web service description framework, to define a web service discovery framework, and to provide scalable web service mediation.

- **VICODI: Visual Contextualisation of Digital Content**.
- **VISION: Next Generation Knowledge Management, http://wim.fzi.de/vision/**. The VISION project will provide a strategic roadmap towards next-generation organisational knowledge management.
- **WIDE: Semantic-Web-Based Information Management and Knowledge-Sharing for Innovative Product Design and Engineering, http://www.cefriel.it/topics/research/default.xml?id=75.** The WIDE project will develop and demonstrate a Semantic Web based information management and knowledge sharing support system for use by teams of industrial designers and product engineers.
- **WISPER: Worldwide Intelligent Semantic Patent Extraction & Retrieval.**
- **WonderWeb: Ontology Infrastructure for the Semantic Web, http://wonderweb.semanticweb.org/.** The aim of the project is to develop the infrastructure required for the large-scale deployment of Ontologies as the foundation for the Semantic Web. This will involve not only the establishment of a Web standard Ontology language, but also the parallel development of the ontological engineering technology that will be required in order to “bring the web to its full potential”.

2 Lessons Learned From the Case Studies

We will discuss lessons learnt in terms of the Swiss Life case study, in terms of the usage of our Ontology language, and in terms of the methodology.

2.1 Debriefing of the Swiss Life Case studies

2.1.1 Complexity of ontologies needed

According to the insight we have gained at Swiss Life during our participation in the Ontoknowledge project Knowledge Management applications typically need ontology languages with a medium degree of expressiveness, reaching from simple taxonomies to hierarchies of concepts with properties and semantic relationships between concepts, including simple cardinality constraints. The more complex ontologies mainly result from the integration with other application systems. For example, the Skills Management application (SkiM) only uses a simple taxonomy for the available skills. However, to realize the application logic a more complex ontology lies underneath, although the user will never see it (cf. **Error! Reference source not found.** below). An extension of SkiM to cover functionalities like gap analysis, the integration with career planning and keeping track of attended training courses would require an additional degree of detail in the ontology.

In contrast, the IAS application only employs an ontology with associative links between concepts and has only an implicit, rudimentary concept hierarchy (via substring matching between two noun groups, like “insurance” and “annuity insurance”). The IAS application shows that an ontology-based application can generate considerable benefit even with a very simple ontology.

2.1.2 Lessons learned from applying the methodology

We experienced that the creation of an ontology is in its early phases a brainstorming activity and not a pure engineering task. Hence, brainstorming tools (e.g. like the used Mind Manager) are valuable add-ons that support the early stages of ontology engineering. The main purpose of these tools is to support the quick and intuitive capturing of knowledge. Especially domain experts not familiar with modelling issues felt comfortable with this tool, in fact they even suggested to use the Mind Manager due to the fact that they were familiar with it. However, at a certain point the process gets less and less brainstorming-driven and more construction-oriented. At Swiss Life, ontology modelling was done directly by a group of domain experts who have been guided and supported by experts in ontology design. In our experience, the domain expert become lost how to structure the ontology when it gets too big or has been changed around too often, because then they loose their feeling of directedness. It becomes more and more difficult to decide where to put which concepts and how to divide the ontology into substructures.

The (not surprising) conclusion is that ontology modelling is a design task which requires certain skills of its own. Ideally, the domain experts have those skills themselves. As such experts rarely exist (yet) the only way to go is to support domain experts by people who have those skills (as we did at Swiss Life). However, the interaction between the domain experts and the ontology design experts is difficult because there is nearly no common ground upon which they can build a common understanding – the domain experts do not understand the design issues, the design experts do not understand the domain (in Swiss Life we had the lucky situation that one domain was IT where the design experts where also domain experts).

```
class-def Department
    instance-of ITDept Department
class-def Skills
    slot-constraint SkillsLevel cardinality 1
slot-def HasSkills
    domain Employee
    range Skills
slot-def WorksInProject
    domain Employee
    range Project
    inverse ProjectMembers
class-def defined ITProject
    subclass-of Project
    slot-constraint ResponsibleDept has-value ITDept
slot-def ManagementLevel
    domain Employee
    range one-of "member" "head-of-group" "head-of-dept" "CEO"
class-def Publishing
    subclass-of Skills
class-def DocumentProcessing
    subclass-of Skills
class-def DesktopPublishing
    subclass-of Publishing and DocumentProcessing
instance-of GeorgeMiller Employee
related HasSkills GeorgeMiller DesktopPublishing3
instance-of DesktopPublishing3 DesktopPublishing
related SkillsLevel DesktopPublishing3 3
```

Fig. 2 Fragment of the ontology underlying the SkiM system but not visible to the user.

We dealt with this situation by upskilling the domain experts with some basic modelling knowledge. We held several workshops, in the beginning to train them and then to support them during the more advanced modelling phases that were more engineering oriented. Within two weeks they were able to model a first skills hierarchy including over 700 skills from three different domains by themselves. Approximately six more weeks were needed to restructure their initial versions in collaboration with modelling experts into an appropriate hierarchy. This was due to time constraints of the participating domain experts. It has to be mentioned that all domain experts were highly self-motivated which certainly had a significant influence on the results produced.

2.1.3 Evaluation

Swiss Life’s experience within Ontoknowledge shows that one of the most critical success factors of ontology-based (knowledge management) systems is their design according to user needs. Of course, this is true for all kinds of application systems, but for ontology-based systems this issue is of special importance because the quality of the information delivered by the system heavily depends on the effort the user invests as well as on his or her skills in harnessing the power of the ontology. This is different to database queries using a key like contract number or service code.

Thus, an ontology-based system must be designed to make its usage as easy as possible. For example, a preliminary field test we did for the IAS case study showed that in general users are extremely unwilling to invest time into the formulation of a query - be it a carefully selected set of free text terms or a set of ontology concepts selected by browsing the concept hierarchy. The reason for this is that a user is typically in a specific business task or decision situation when he or she decides that certain information is needed to be able to proceed. Although seen as a necessity, it is also perceived as a nuisance, something that must be done but distracts from the real work. In the IAS application we solved this problem by not forcing the user to formulate an elaborated query based on the underlying ontology, but to offer him or her the opportunity to formulate a more sophisticated query after the first query result. If that result is unsatisfactory the user's motivation to put in the extra effort is much higher. This (initial) hypothesis was confirmed by the observed user behavior after the introduction of the system.

In the SkiM application the user does not have much of a chance to obtain a query result at all unless he or she enters terms from the ontology. But still, the SkiM system allows the input of any terms, which might still provide some meaningful result because all non-ontology terms are matched against the free text parts of an employee's skills description (unless explicitly prohibited by the user).

2.2 The lessons learned from the development and use or non-use of the OWL (ex-OIL) language in the project

Compared with the external use and adaptation of OIL, the internal use of OIL within Ontoknowledge in case studies as well as in the developed tools is rather disappointing. None of the case studies really employed OIL. They all refer to RDF Schema or a subset of it which we called **Core OIL** to mask our "failure". What are the reasons for this:

- Some case studies already started before even the first version of OIL was available.
- OIL lacks any tutorial support, any customized tool support, and any real practical experience. From a user point of view this makes it nearly unusable for the moment.
- OIL or OWL are ongoing standardization efforts including high change rates. No user nor tool developer can trust the current language version as being final and not just an intermediate step. Using OIL is of high risk, both for uses in case studies as well as for tool developers.

The lack of actual use of OIL may also be an indication for three different problems related to OIL:

- **OIL is too less expressive.** Many things cannot be expressed in it but could be easily expressed in a rule-based language like F-logic (cf. M. Kifer, G. Lausen, and J. Wu: Logical Foundations of Object-Oriented and Frame-Based La) oriented on reasoning over instances. The initial approaches to the semantic web were oriented around this paradigm).
- **OIL is too expressive.** OIL is logic based and OIL is description logic based. Many people find it difficult or not worth while to express themselves within logic. For example, non of the standard ontologies in electronic commerce or widely used

ontologies such as Wordnet[3] make use of any axiomatic statements. Mostly they are simple taxonomies enriched by attributes in the best case. Description logic adds a specific feature: Concept hierarchies do not need to be defined explicitly by can be defined implicitly by complex definitions of classes and properties. Many people may find it easier to directly define is-a relationships instead of enforcing them by complex and well-thought axiomatic definition of classes and properties.

- **Ontologies should not be based on formal logic.** People with a background in databases wonder in general whether axioms, i.e., complex logical statements should be part of an ontology. They tend to be application specific and very difficult to exchange and reuse in a different context. Spoken frankly, most of our experience conform with this statement. Still we may hope to make the world a better place in the future enabling more than just the exchange of concept taxonomies and some attributes.

In any case, it is too early to take a final conclusion from our exercise. We hope that we will get more insights from the use cases collected by the *Web-Ontology (WebOnt) Working Group*[4] . Also the *SIG on Ontology Language Standards*[5] of *Ontoweb*[6] is supposed to bring further insights here.

2.3 Lessons learnt and recommendations on ontologies use and skills requirements

The main lessons learnt from applying the methodology in the case studies are:

- Early ontology development is often unstructured brainstorming rather then careful design. Such brainstorming is currently unsupported by many ontology engineering tools like Protege or OilEd. We relied on a state-of-the, commercially successful and widely used brainstorming tool. We extended to functionalities of this tool to meet our requirements, in particular we connected the tool with our OntoEdit with the plug-in Mind2Onto to close the gap between capturing and formalization of knowledge.
- Different processes drive KM projects, but „Human Issues“ might dominate other ones (as already outlined by Davenport). We had to learn hard lessons there, two of the case studies were heavily affected by major internal restructurings of the companies.
- Collaborative ontology engineering requires physical presence and advanced tool support. On the one hand we not only had to train, support and guide domain experts, but also to personally motivate them to contribute to our project. On the other hand they could work in the mean time by themselves on “their part of the ontology” due to the strong support of the tools.

2.4 Enersearch Case Study

In the EnerSearch case study we encountered the problem that domain experts in industrial contexts are hard to access. This was especially true since the collaboration with our project was neither part of their job nor enforced by their management. To build up an ontology

[3] http://www.cogsci.princeton.edu/~wn/

[4] http://www.w3.org/2001/sw/WebOnt/

[5] http://www.cs.man.ac.uk/~horrocks/OntoWeb/SIG/

[6] http://www.ontoweb.org/

manually like in the Swiss Life case study was therefore not possible. We therefore followed the approach of using a lightweight ontology which was extracted manually out of the given document base.

2.4.1 Lessons learnt from the EnerSearch Virtual Organization case study

This section summarizes the lessons learned through using On-To-Knowledge tools, technologies and methodology for creating an ontology-based content management system for the EnerSearch research organization. We also include the main conclusions from the end-user evaluation of the ontology-based search and browsing interfaces of the EnerSearch system.

As there have been very few projects before that would have ventured into providing an integrated ontology-based solution we expect that these experiences will provide significant help for the implementation and assessment of future ontology-based designs. For a more detailed description of the EnerSearch scenario we refer the reader to [On-To-Knowledge del. 27, 28]. We point to [On-To-Knowledge del. 29] for details on technological and end-user evaluation.

2.4.2 Ontology engineering

As in the SwissLife IAS case study, EnerSearch has chosen an automated method for ontology extraction. This decision, however, was based on the specific case of EnerSearch: due its nature as Virtual Organization access to domain experts was restricted and they were not expected to be actively involved in creating and updating the domain ontology. The extent of the domain in which the organization operates and the corresponding size of the domain corpora have also contributed to this choice. The low cost of ontology extraction compared to manual authoring has been a significant factor in favor of automated methods.

We also have to note that for reasons similar to the ones mentioned above (size of the domain, non-technical, loosely affiliated users etc.) it's also expected that automated ontology acquisition techniques will dominate the future Semantic Web. Therefore ontology extraction was a challenge that was considered to be worth pursuing from an applications perspective.

Ontology extraction using OntoExtract was a fairly straightforward and speedy method. The resulting ontology, however, lacked a solid class hierarchy, property values and the so-called cross-taxonomical relationships were also hard to interpret at times. It was also noted that the amount and extent of overlaps between the resulting ontological classes was unusually high.

These problems are unlikely to be solved without the addition of background knowledge. Building a class hierarchy, for example, involves modeling decisions based on the domain in question, which OntoExtract is not aware of. With the addition of background knowledge about the semantic distances of concepts, OntoExtract could also potentially consolidate some of the ontological classes into larger clusters. (OntoExtract currently features only a limited background knowledge facility, which is used for giving a concrete name to the generic oe:hasSomeProperty relations.)

On the positive side, OntoExtract enriched the documents with significant metadata (key concepts, summary). This allowed us to differentiate between concepts and build a browsing and search interface that is still navigable for the user. This largely corresponds to the findings of the SwissLife case: while automated methods result in extremely lightweight ontologies (an expected outcome), innovate methods to ontology-based information retrieval can still lead to a satisfying user experience.

The EnerSearch case also coincided with the SwissLife experience in that more expressive constructs entered the ontology through integration with existing systems. In this case this integration was the reengineering of the publication database into an ontology format. This ontology contained metainformation about publications such as names of authors, date of publication etc.

The ontology obtained through natural language processing (NLP) has been represented in RDF(S). The use of OIL (beyond OIL Core) was ruled out as the case study could not use the additional expressiveness of the language, due to (1) the application domain and (2) the methods we have chosen. Moreover, the case study application would have suffered from the computational complexity of a more expressive language.

On the other hand, RDF(S) lacked some of the features that would have been required, such as the ability to express equality between resources. (While many believed that a similar effect could be achieved by cycles of subClassOf/subPropertyOf relations, this has been disproved.) This has also been one of the major stumbling blocks for realizing an iterative process of ontology engineering that would have combined automated extraction with manual efforts. (Without an ability to cluster concepts, the size of the extracted ontology ruled out manual approaches). Drawing attention to the gravity of this omission was one of the major outcomes of the case study from a technological perspective.

The standardization body working on the future OWL has also acknowledged the need for a less expressive language for such domains as content management. Plans are now in place for OWL Lite, a language that would provide the limited set of primitives (but still including equality) required by lightweight applications.

2.4.3 Ontology storage, query, transformation

Sesame, the RDF(S) storage and query facility of the project was used to store the resulting ontology, to perform the standard inferencing prescribed by the Model Theory and to evaluate simple queries over the data. The OMM and BOR extensions were not used as their development came late in the project. (However, OMM was tested for compatibility with the EnerSearch application.) Also, the BOR reasoner was not applicable in the case study scenario as discussed above.

While Sesame has proved to be very stable and mature from the very beginning, the performance and scalability of the facility was put to the test by the EnerSearch application. (Due to the automated extraction, the number of resources went far beyond of what is typical for manually authored ontologies.) We also concluded that performance comes to play a crucial role when integrating Sesame into a larger system. As a direct response, the developers of Sesame have significantly improved the performance of the inference engine and the query evaluation subsystem, the most computational intensive parts of the product. (The query language RQL itself is also under revision.) We expect in the near future further dramatic improvements from an embedded version of Sesame or a SAIL implementation that would carry out these operations on data stored in main memory.

We also found that client-side operations are necessary for reasons other than performance, such as extending the expressiveness of representation or carrying out the inferences based on client-side semantics. Therefore we reconsidered the distribution of data and operations between client(s) and server and developed a package for client-side query and transformations on top of the Sesame Client API. Among others this enabled us to scale up our applications (the Spectacle client in particular) to the size of the EnerSearch domain and to tailor our ontology to the needs of the retrieval tools. All in all this division provides a more sensible architecture for future applications.

2.4.4 Ontology-based information retrieval

Our experiment involved implementing two different approaches to information retrieval - searching and browsing-, using the ontology we obtained through combining the domain ontology and the ontology obtained by converting the EnerSearch publication database. In both cases, however, simple transformations were necessary to achieve the best user experience.

In the case of QuizRDF this meant filtering out non-essential concepts and carrying out a simple transformation from class labels to property values. Once these had been executed, applying the QuizRDF search engine generator was a simple, one-step process. On the downside, this automatically generated search interface could not be customized to reflect the semantics of the data, i.e. the facility is built on the minimal semantics of RDF(S) constructs. For example, the interface offers browsing over the sub- and superclasses of concepts, but has no notion of class-level relationships, therefore it's not possible to navigate on related classes.

We should also note that QuizRDF performs best on ontologies where classes have a rich set of properties that can be used as search criteria. This was not the case in the EnerSearch ontology, where only one of the classes has relations on the instance level.

The Spectacle presentation platform provided large freedom for the system engineer in tailoring the browsing interface to the ontology. However, unlike in the case of QuizRDF, creating the interface was a sizeable programming task. Using Spectacle also involved learning the methodology behind the product.

In our esteem the investment in learning and high-tech engineering is paid back when it comes to automated generation of content. A particularly convincing feature of this phase is the ability to apply different heuristics to cut back navigation paths that are meaningless or would not result in efficient navigation. The result is a website that is not only flexible in the navigation options it offers, but efficient in guiding the user as well.

One particular challenge we met in building the EnerSearch presentation was due to the peculiar nature of the ontology: a large amount of concepts without a solid class hierarchy and extensive overlaps between concepts. The lack of a firm class hierarchy was compensated by a careful choice for the mapping from ontology to navigation structure. With a hierarchical access method it was still possible to create the appearance of a hierarchy based on intersection semantics. The price to pay was the number of sections that resulted, which threatened with yet another scalability problem. The solution came from the developers of Spectacle who made it possible to generate sections dynamically, i.e. on demand. This means that a Spectacle page now only needs to rendered when the first user tries to access it.

2.4.5 On-To-Knowledge Methodology

Through the feasibility study for the ontology-based system, we concluded that the business case for ontology-based search and navigation is particularly strong for virtual enterprises, such as EnerSearch, whose main value driver is the creation and dissemination of (scientific) knowledge. For this kind of enterprise, the gains from employing ontologies can offset the significant technological risks involved with using advanced semantic technologies.

The distributed fashion of the organization also determines the kind of ontologies and applications to be employed. In this setting low-threshold solutions such as ontology-based browsing are preferred over applications that require a larger commitment, such as the OntoShare collaboration environment, which has been also proposed in On-To-Knowledge. Also, the ontology underlying virtual enterprise applications are likely to be lightweight and developed in a distributed fashion, as formalization requires central control to avoid overlaps and inconsistencies in engineering. Note that we also believe – contrary to previous methodological assumptions- that the degree of formality of an ontology is not necessarily a measure of quality in such environment. (As discussed above, we have also seen that lightweight ontologies provide a fair trade-off in terms of efficiency vs. cost.)

The case study validated the On-To-Knowledge approach of providing a set of tools and technologies for building customized knowledge management solutions, as opposed to providing a one-size-fits-all knowledge system. Tool integration, however, required the development of additional components and raised performance/scalability issues that were not apparent by judging the tools individually. Some of these issues have even led us to reconsider the division of work between components. Thus, the evaluation of ontology-based tools, as discussed in the methodology, was extended to cover not only standalone tools, but also the properties of various architectures or frameworks of ontology-based solutions. Our evidence suggests that (1) there are properties of such combinations (such as the scalability and performance of the entire architecture) that cannot be estimated by judging individual tools and services and (2) there may become visible a need for additional software when composing tools, e.g. for transformations and mapping. Investigating these challenges will be especially important in the next stage of Semantic Web developments when the focus from individual tools will shift to combining tools or services.

2.4.6 End-User Evaluation

In this part we discuss the lessons learnt form the end-user evaluation. The information is based on the findings from the two case studies carried out by SwissLife and EnerSearch.

Design considerations: In the SwissLife's case study as well as for the EnerSearch case study, shows that one of the most critical success factors of ontology-based systems is the design according to the user needs. This especially important for ontology based systems since there is a clear trade-off between the user and the system. The quality of the information that extracts from the tools is dependent on the effort the users spend in using the tools. Further it is important to consider that an ontology-based system must be designed to make its usage as easy as possible.

In the case of EnerSearch the involvement of knowledge users in the experiment at an early stage was important. This because of the interaction between the users and tasks on the one hand, and their pre-knowledge, or lack thereof, of domain and/or systems on the other hand. It was also the fact that we wanted to evaluate different types of elements:

(1) Different modes of information gathering that were employed by users when trying to find the information they were looking for. If we want to evaluate semantic methods we first have to establish a clear reference point for comparison, or some kind of unit in terms of which we can measure the quality of new ways of doing things. Two different semantic web-style information modes were tested - ontology-based search and ontology-based browsing (compared with existing keyword-based search tool).

(2) Different target user groups that were to be distinguished from the viewpoint of differences in interest, motivation, the nature of information they pursued, and background knowledge they brought.

(3) Different personal styles of information used that related to differences in individual cognitive styles in handling information. In general it is highly important to carry out a proper segmentation of target users, because often they are quite different, each bringing with them their own view or, so to speak, "ontology" regarding the information they seek. These differences will clearly influence their perception of how good semantic web methods works.

The SwissLife IAS case study showed that users in general are unwilling to invest time in query for information. The reason for this is that users is typically in a specific business task or decision situation when he or she decides that certain information is needed to proceed. The similar phenomenon was noticed in the EnerSearch case study. Several types of target users were identified but far from all were actually part of the case study in the end. The main reason for the decrease in numbers of test users was the long time it took to finish the test and in the fact that in this kind of experiment it is always a certain number that drops out in the for different other reasons.

Requirements for experiment: Semantic Web tool tests and case studies in the field require a very careful experiment design. Prospective test user groups and test tasks must be carefully balanced to allow for adequate empirical-statistical testing of hypotheses that must be explicitly formulated in advance. Empirical data gathering in such experiments must be rich, including various qualitative methods – such as pre- and post-trial semi-open interviews, collecting verbal protocols during the experiment, onsite observation – as well as quantitative methods – for example questionnaires, electronic logging of actions and execution times, and statistical processing of resulting data.

In the EnerSearch case study we focused on testing four different hypotheses:

(1) Users will be able to complete information-finding tasks in less time using the ontology-based semantic access tools than with the current mainstream keyword-based free text search.
(2) Users will make fewer mistakes during a search task using the ontology-based semantic access tools than with the current mainstream keyword-based free text search.
(3) The ontology-based tools will be experienced as more difficult to use and number of mistakes will be more noticeable for less experienced users of the EnerSearch database.
(4) The ontology-based semantic access tools will be perceived differently according to what type of decision style you have as a person in the context of knowledge-acquisition scenarios.

Requirements for tools: The ontology-based tools gives the user a good picture of the inherit data with concepts but the major problem is that whenever a document was found related to the specific question there were problems finding the answer within the document. Action to be taken was to make a pointer within the documents that could point out the relevant parts in the document.

Creating relevant questions was not an easy task. It was necessary to make sure that the tools (both the two ontology based and the basic search tool) questions were linked to the search tools. We didn't want to make the question either too hard or too easy to answer at the same time we wanted to use the advantages with the different search tools.

Two major issues that the users expected from the ontology based tools were that it should be faster and easier to find information within the ontology and at the same time receive reliable information in a logical way. The importance of selecting ”right” concepts was of great importance otherwise it would not work. Another issue concerning the usability of the ontology-based tools was that there should not be to many clicks before the user receives a satisfying answer to a specific question.

The lesson learnt from SwissLife was that in the SkiM application the user didn't have the opportunity to obtain a query result unless the terms from the ontology was used. But still, the user had the possibility to use any search term, which still provided meaningful results due to the fact that all non-ontology terms were matched against the free text parts of an employee's skills description.

3 Dissemination

Ontoknowledge is one of the most highly visible research projects in the semantic web area. This success has been based on a large number of dissemination activities. Here we will only mention the major ones.

We founded a thematic network called ***Ontoweb***[7] with around 150 members. Setting up a thematic network seemed to be a proper dissemination activity both in terms of visibility and travelling budget. The goal of the OntoWeb Network is

- to bring together researchers and industrials working on Ontologies and semantic web technology,
- promoting interdisciplinary work and
- strengthening the European influence on standardization efforts.

We helped to create the ***1st International Semantic Web Conference*** and we will organize the ***2nd*** one in October 2003 in Florida,U.S.A.[8]

Fig. 3 International Semantic Web Conference.

We are strongly involved in creating an international standard called ***OWL***. Within Ontoknowledge we developed a web-based layered framework of ontology representation languages called OIL. In co-operation with the large US project DAML[9] we developed a joined consensual sub-dialect called DAML+OIL. The semantic web working group of the W3C uses DAML+OIL as departure point for their standardization effort. We are in the

[7] http://www.ontoweb.org/

[8] http://iswc.semanticweb.org/

[9] http://www.daml.org/

process of publishing OWL as a W3C recommendation.

Sesame has high international visibility. It is one of the few world-wide RDFS Repositories and developed as open source.[10] It is now funded by a non-profit organization called Nlnet.[11] Here are some figures on Sesame downloads and use on Sesame current version: 0.6.1 (released September 6):

- sourceforge.net software downloads (stats by Sourceforge):
  since release 0.6.1 (Sep 6): 69
  since start of OS project (Mar 14): 798
- sesame.aidministrator.nl visitors (stats by Webalizer v2.01)
  visitors August only: 2764
  avg. visitors per day August only: 89

Interesting side note: the Sesame Open Source project currently ranks #8 (out of 581) in the listing of most active AI projects on Sourceforge. To compare: the OpenCyc project is listed as #7. See http://sourceforge.net/softwaremap/trove_list.php?form_cat=133.

Finally, the Ontoknowledge book) and the web site *http://www.ontoknowledge.org/* complete our dissemination
strategies.

[10]http://sourceforge.net/projects/sesame/

[11]http://www.nlnet.nl/

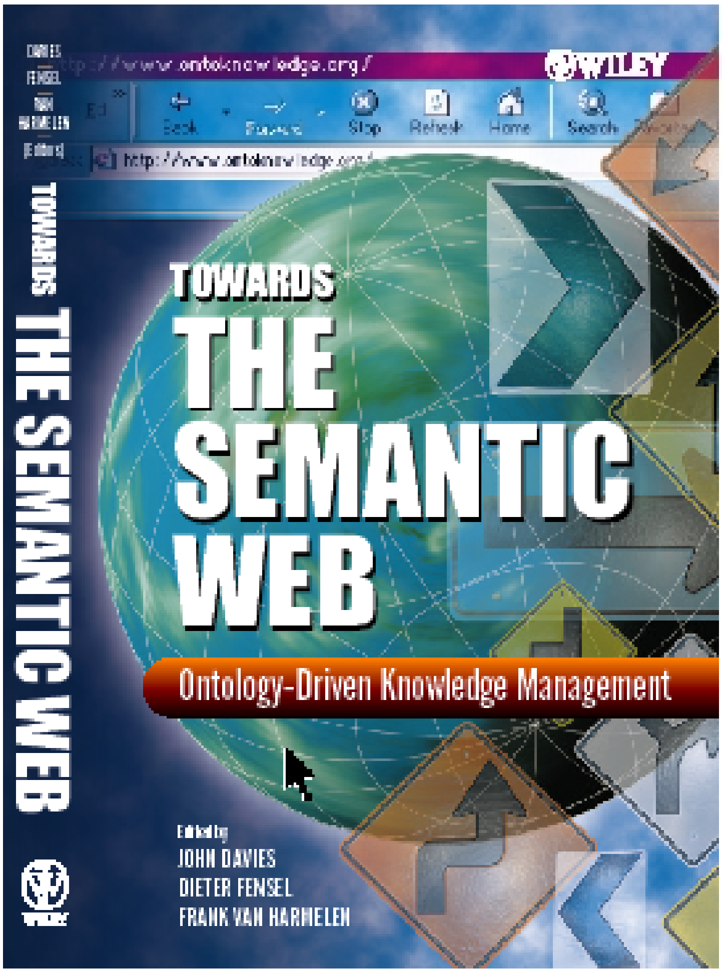

Fig. 4 The Ontoknowledge book.

4 Exploitations

The Exploitation Management Board met for the final time in September 2002 to finales the consortium's approach to exploitation and to put in place suitable agreements to allow commercial discussions to continue following the end of the project. An detailed exploitation report has been produced. The report begins by reviewing the market which Ontoknowledge addresses. It then considers the Ontoknowledge software tools and architecture and the extent to which they address identified segments of the market. A range of options for exploitation are then reviewed and the consortium's preferred options are set out. The report then describes the consortium chosen mechanism for furthering its exploitation goals, namely a Memorandum of Understanding which provides a framework for 2 strands of the consortium's plans for exploitation. These 2 strands are as follows: the productisation of the entire Ontoknowledge toolset (along with the accompanying OTK KM methodology) by one partner on the one hand and a reselling agreement with respect to individuals software tools between partners on the other. Each partner's individual exploitation plans in terms of markets, competitors, commercial models, successes to date and commercial forecasts are then described. Finally, the SWOT analysis produced in the Ontoknowledge Market and

Competitor Analysis is revisited: the extent to which the earlier perceived weaknesses have been addressed is assessed and some new aspects of the SWOT analysis are identified and discussed.

One of the involved commercial partners, Sirma AI, Ltd. - Artificial Intelligence Labs, OntoText Lab, will become the exploitation partner for the overall tool suite. Sirma joined Ontoknowledge on a later stage. It was enabled by an additional funding line for partners from newly associated states. We congratulate the European Commission for this initiative because the involvement of Sirma in Ontoknowledge significantly improved our overall results. Extremely knowledgeable, motivated (and low paid) IT experts from Bulgaria helped to integrate the overall tool suite and provided significant contributions such as the Ontology Management Module (OMM).

5 Summary

We are to close to the picture to provide an adequate summary of our achievements. Therefore we decided to let the reviewers speak:

> "*On the bottom line, OTK (On-To-Knowledge) has resulted in a great academic leap with a set of potential commercial applications; the marketing future for an Ontology Suite, i.e. the translation of the architecture into products, however remains doubtful in the given project exploitation scenarios.*
>
> *As the project stands at its end, the OTK results include an elegantly layered application architecture, a useful and practical methodology, a standards triggering language (OWL), a demonstrable tool set and three case studies of varying depth. Considering the very nature of this endeavour, which is dealing with pace making technologies, OTK is a fine achievement. In the technology race OTK has a clear lead, though rapidly decreasing, above a still fragmented knowledge management (KM) market portfolio.*
>
> *The consortium negotiated this unknown terrain in a pragmatic fashion sticking to the original project scope. It is actually this attitude that produced the most noticeable results:*
>
> - *a clear cut conceptual architecture (integrating the tool set)*
> - *a useful series of tools (as shown in the case studies)*
> - *a practical methodology for introducing KM concepts and tools into enterprises.*
>
> *By doing so, OTK ´s direction evokes a process that is much closer to industrial R&D than classical applied science. But, alas for the project, the material world of business demands a reasonable return on investment and here is a twist: in the true notion of a product lifecycle OTK represents only an early laboratory prototype with some convincing demonstrator capability.*"
>
> [Final Review Report]

6 References

[Davies et al., 2002(a)]
J. Davies, U. Krohn, and R. Weeks: QuizRDF: search technology for the semantic web. In WWW2002 workshop on RDF & Semantic Web Applications, 11th International WWW Conference WWW2002, Hawaii, USA, 2002.

[Davies et al., 2002(b)]
J. Davies, A. Duke, and A. Stonkus: OntoShare: Using Ontologies for Knowledge Sharing. In Proceedings of the WWW2002 Semantic Web workshop, 11th International WWW Conference WWW2002, Hawaii, USA, 2002.

[Davies et al., 2002(c)]
J. Davies, D. Fensel, and F. van Harmelen (eds.): The Semantic Web: Ontology-driven Knowledge Management, John Wiley, 2002.

[Fensel, 2001]
D. Fensel: Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag, Berlin, 2001.

[Fensel et al., 1998]
D. Fensel, S. Decker, M. Erdmann, and R. Studer: Ontobroker: The Very High Idea. In *Proceedings of the 11th International Flairs Conference (FLAIRS-98)*, Sanibal Island, Florida, USA, 131-135, May 1998.

[Fensel et al., to 2001]
D. Fensel, I. Horrocks, F. van Harmelen, D. McGuinness, and P. F. Patel-Schneider: OIL: Ontology Infrastructure to Enable the Semantic Web, IEEE Intelligent System, 16(2), 2001.

[Kifer et al., 1995]
M. Kifer, G. Lausen, and J. Wu: Logical Foundations of Object-Oriented and Frame-Based Languages, *Journal of the ACM*, 42, 1995.

[Luke et al., 1996]
S. Luke, L. Spector, D. Rager, and J. Hendler: Ontology-based Web Agents. In *Proceedings of First International Conference on Autonomous Agents 1997, AA-97*.

[Staab et al., 2001]
S. Staab, H.-P. Schnurr, R. Studer, and Y. Sure: Knowledge Processes and Ontologies, IEEE Intelligent Systems, 16(1), 2001.

[Sure et al., 2002a]
Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer and D. Wenke: OntoEdit: Collaborative Ontology Engineering for the Semantic Web. In Proceedings of the first International Semantic Web Conference 2002 (ISWC 2002), June 9-12 2002, Sardinia, Italy.

[Sure et al., 2002b]
Y. Sure, S. Staab, J. Angele. OntoEdit: Guiding Ontology Development by Methodology and Inferencing. In: R. Meersman, Z. Tari et al. (eds.). Proceedings of the Confederated International Conferences CoopIS, DOA and ODBASE 2002, October 28th - November 1st, 2002, University of California, Irvine, USA, Springer, LNCS 2519, pages 1205-1222.

7 Annex

We provide the list of all deliverables, some of the related papers, and some of the given dissemination talks.

7.1 Deliverables of the Project (acessible via the web site)

Deliverable 0: OIL: A Standard Proposal for The Semantic Web
Deliverable 1: Ontology Language Version 1
Deliverable 2: Ontology Language v.2.
Deliverable 3: OntoEdit
Deliverable 4: Ontology Management
Deliverable 5: Information Extraction
Deliverable 6: Ontology Extraction Tool
Deliverable 7: O-Wrapper
Deliverable 8: Representation and Query Languages for Semistructured Data - State of the art report
Deliverable 9: Query Language Definition
Deliverable 10: Query Engine
Deliverable 11: Search Facility
Deliverable 12: Knowledge Sharing Facility
Deliverable 13: Visualization Facility
Deliverable 14: User Profile Construction
Deliverable 15: Ontoknowledge Methodology
Deliverable 16: Ontoknowledge Methodology - Employed and Evaluated Version
Deliverable 17: Methodology Expansion
Deliverable 18: KM-Methodology
Deliverable 19: Swiss Life Application Study
Deliverable 20: Organizational Memory: Description of Case Study Prototypes
Deliverable 21: OM Prototype Evaluation Document
Deliverable 22: OM-Application
Deliverable 24: Requirement Analysis Document
Deliverable 25: Help Desk Prototype
Deliverable 26: Evaluation Document
Deliverable 27: Ontoknowledge EnerSearch Virtual Organization Case Study: Requirements Analysis Document
Deliverable 28: VE Prototype
Deliverable 29: Evaluation Document
Deliverable 30: Ontoknowledge Web Page
Deliverable 31: Intermediate Workshop
Deliverable 32: Final Workshop
Deliverable 33: Project Presentation on Ontoknowledge: Content-driven Knowledge Management Tools through Evolving Ontologies
Deliverable 34: Dissemination and Use Plan
Deliverable 35: Technology Assessment
Deliverable 36: Draft Exploitation Plan
Deliverable 37: Market and Competitor Report
Deliverable 38: Ontology Middleware: Analysis and Design
Deliverable 39: OMM Development

Deliverable 40: LINRO analysis and prototype
Deliverable 41: OMM Integration
Deliverable 42: Final Exploitation Plan

Deliverable 43: Final Project Report
Extra Deliverable: Interoperability and scalability of Ontoknowledge tools

7.2 Further Publications of the Ontoknowledge Project

1. P. Patel-Schneider and I. Horrocks and F. van Harmelen, Reviewing the Design of DAML+OIL: Proceedings of the Eighteenth National Conference on 2002, R. Dechter and M. Kearns and R. Sutton,, July.

2. S. Decker and D. Fensel and F. van Harmelen and I. Horrocks and S. Melnik and M. Klein and J. Broekstra, Knowledge Representation on the Web, F. Baader, International Workshop on Description Logics (DL'00), 2000,

3. D. Fensel and M. Cruzeby and F. van Harmelen and I. Horrocks, OIL and UPML: a unifying framework for the Knowledge Web, Proceedings of the ECAI'00 workshop on applications of ontologies and problem-solving methods, 2000, Berlin, August,

4. Michel Klein and Dieter Fensel and Frank van Harmelen and Ian Horrocks, The relation between ontologies and schema-languages: Translating OIL-specifications in XML-schema, Proceedings of the ECAI'00 workshop on applications of ontologies and problem-solving methods, 2000, Berlin, August.

5. D. Fensel and I. Horrocks and F. Van Harmelen and S. Decker and M. Erdmann and M. Klein, OIL in a nutshell, R. Dieng, 1937, Lecture Notes in Artificial Intelligence, 1-16, Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW'00), Springer-Verlag, 2000.

6. Heiner Stuckenschmidt and Jens Hartmann and Frank van Harmelen, Learning Structural Classification Rules for Web-page Categorization, Proceedings of the Fifteenth International FLAIRS conference, 2002, S. Haller and G. Simmons, May.

7. Dieter Fensel and Frank van Harmelen and Ying Ding and Michel Klein and Hans Akkermans and Jeen Broekstra and Arjohn Kampman and Jos van der Meer and York Sure and Rudi Studer and Uwe Krohn and John Davies and Robert Engels and Victor Iosif and Atanas Kiryakov and Thorsten Lau and Ulrich Reimer and Ian Horrocks, Ontoknowledge in a Nutshell, IEEE Computer, 2002.

8. Stefan Decker and Sergey Melnik and Frank Van Harmelen and Dieter Fensel and Michel Klein and Jeen Broekstra and Michael Erdmann and Ian Horrocks, The Semantic Web: The roles of XML and RDF, IEEE Internet Computing, 2000, 15, 3, 63--74, October.

9. Frank Van Harmelen and Ian Horrocks, Questions and answers OIL, IEEE Intelligent Systems, 2000, 15, 6, 69--72.

10. Dieter Fensel and Ian Horrocks and Frank van Harmelen and Deborah L. McGuinness and Peter F. Patel-Schneider, OIL: An Ontology Infrastructure for the Semantic Web, IEEE Intelligent Systems, 2001, 16, 2.

11. Frank van Harmelen, The complexity of the Web ontology language, IEEE Intelligent Systems, 2002, 17, March/April.

12. F. van Harmelen and D. Fensel, Practical Knowledge Representation for the Web, D. Fensel, Proceedings of the IJCAI'99 Workshop on Intelligent Information Integration, 1999.

13. Jeen Broekstra and Arjohn Kampman and Frank van Harmelen, Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema, Proceedings of the First Internation Semantic Web Conference, 2002, 54--68, I. Horrocks and J. Hendler, July, Springer Verlag, Lecture Notes in Computer Science, 2342.

14. H. Stuckenschmidt and F. van Harmelen, Ontology-based Metadata Generation from Semi-Structured Information, Proceedings of the First Conference on Knowledge Capture (K-CAP'01), 2001, Victoria, Canada, October.

15. Dieter Fensel and Enrico Motta and V. Richard Benjamins and Monica Crubezy and Stefan Decker and Mauro Gaspari and Rix Groenboom and William Grosso and Frank van Harmelen and Mark Musen and Enric Plaza and Guus Schreiber and Rudi Studer and Bob Wielinga, The Unified Problem-solving Method Development Language UPML, Knowledge and Information Systems, 2002.

16. F. van Harmelen, How the Semantic Web will change KR: challenges and opportunities for a new research agenda, The Knowledge Engineering Review, 17, 1, 2002.

17. J. Broekstra and A. Kampman and F. van Harmelen, Sesame: An Architecture for Storing and Querying RDF Data and Schema Information , Semantics for the WWW, MIT Press, 2001, D. Fensel, J. Hendler, H. Lieberman and W. Wahlster.

18. Frank van Harmelen and Arjohn Kampman and Heiner Stuckenschmidt and Thomas Vogele, Knowledge-Based Meta-Data Validation: Analyzing a Web-Based Information System, Fourtheenth International Symposium Informatics for Environmental Protection, German Computer Society, K. Greve, 2000.

19. Frank van Harmelen and Jeen Broekstra and Christiaan Fluit and Herko ter Horst and Arjohn Kampman and Jos van der Meer and Marta Sabou, Ontology-based Information Visualisation, Proceedings of the workshop on Visualisation of the Semantic Web (VSW'01), 2001, London, in conjunction with the 5th International Conference on Information Visualisation, September.

20. Christiaan Fluit and Marta Sabou and Frank van Harmelen, Ontology-based Information Visualisation, Visualising the Semantic Web, 2002, Springer Verlag, Vladimir Geroimenko.

21. H. Stuckenschmidt and F. van Harmelen, Knowledge-Based Validation, Aggregation and Visualization of Meta-Data: Analyzing a Web-Based Information System Proceedings of the first Asia-Pacific conference on Web Intelligence (WI'2001), 2001, N. Zhong and Y. Yao, 2198, 217--226, Springer-Verlag, Lecture Notes in Artificial Intelligence.

22. J. Broekstra and M. Klein and S. Decker and D. Fensel and F. van Harmelen and I. Horrocks, Enabling knowledge representation on the Web by Extending RDF Schema, Proceedings of the tenth World Wide Web conference WWWW'10, 2001, Hong Kong, May.

23. Dieter Fensel and Frank van Harmelen and Michel Klein and Hans Akkermans and Jeen Broekstra and Christiaan Fluit and Jos van der Meer and Hans-Peter Schnurr and Rudi Studer and John Hughes and Uwe Krohn and John Davies and Robert Engels and Bernt Bremdal and Fredrik Ygge and Thorsten Lau and Bernd Novotny and Ulrich Reimer and Ian Horrocks, Ontoknowledge: Ontology-based Tools for Knowledge Management, eBusiness and eWork, 2000, Madrid, October.

24. R.Engels and B.A. Bremdal.CORPORUM: A Workbench for the Semantic Web. Semantic Web Mining workshop. PKDD/ECML - 01. Freiburg, Germany, 2001.

25. R. R. Jones, B. A. Bremdal, Ch. Spaggiari, F. Johansen and R. Engels. Knowledge Management through Content Interpretation. AISC2000, September 21th-22th, Hong Kong, China, 2000.

26. Y. Sure, S. Staab, J. Angele. OntoEdit: Guiding Ontology Development by Methodology and Inferencing. In: Proceedings of the International Conference on Ontologies, Databases and Applications of SEmantics ODBASE 2002, October 28 - November 1, 2002, University of California, Irvine, USA, Springer, LNCS.

27. V. Iosif and Y. Sure. Exploring potential Benefits of the Semantic Web for Virtual Organizations. Submitted 2002.

28. Th. Lau and Y. Sure. Introducing Ontology-based Skills Management at a large Insurance Company. In: Modellierung 2002, Modellierung in der Praxis - Modellierung für die Praxis, Tutzing, Deutschland, 25.-27. März 2002.

29. Y. Sure. Ontoknowledge -- Ontology based Knowledge Management Tools and their Application. In: German Journal Künstliche Intelligenz, January 2002 (1/02), Special Issue on Knowledge Management.

30. Y. Sure et al.. Ontoknowledge: Semantic Web enabled Knowledge Management. In: Web Intelligence. Ning Zhong (ed.), Springer, to appear 2002.

31. S. Staab, R. Studer and Y. Sure. Knowledge Processes and Meta Processes in Ontology-based Knowledge Management. In: Handbook on Knowledge Management. C. W. Holsapple (ed.), Springer, to appear 2002.

32. Y. Sure. A Tool-supported Methodology for Ontology-based Knowledge Management. In: The Ontology and Modeling of Real Estate Transactions in European Juristictions. E. Stubkjaer (ed.), International Land Management Series, Ashgate, to appear 2002.

33. Atanas Kiryakov and Damyan Ognyanov. Tracking Changes in RDF(S) Repositories In the Proceedings of Workshop on Knowledge Transformation for the Semantic Web, at the 15th ECAI, pages 27-35. Lyon, France, July 23, 2002.

7.3 Some Dissemination Talks

1 Several invited talks in Norway on EU infodays. Dissemination on OntoKnowledge and its results. Those days where organised by institutes like SND, Norsk Forksningsråd, European Information Centres, and Agder Forskning.

2 May'02: Organising an advanced course on Semantic Web for the Dutch Graduate School on Knowledge- and Information Systems.

3 Organised a masterclass on the Semantic Web with Pat Hayes and Peter Patel-Schneider as speakers (and over a hundred people attending).

4 Industrial seminar on Semantic Web at Elsevier Science in Amsterdam on June 24.

5 Masterclass on Semantic Web for Dutch W3C membership in Den Haag, summer '01.

6 Industrial seminar on Semantic Web at Philips Research Labs Eindhoven, summer '01.

7 Invited talk for Dec'01 meeting of AgentLinks (EU Network of Excellence).

8 Keynote speach at Dutch IT Conference organised by Dutch Ministry of Economic Affairs.

9 Combined invited speaker at Distributed Objects and Applications (DOA´02), Ontologies, Databases, and Applications of Semantics (ODBASE´02), and Cooperative Information Systems (CoopIS´02), October & November 2002.

10 Keynote speaker at the XML Finland 2002: Towards the Semantic Web and Web Services, Helsinki, Finland, October 2002.

11 Invited speaker at the 25th German conference on Artificial Intelligence, Aachen, September 2002.

12 Invited speaker at IBM Watson, New York, September 2002.

13 Invited Speaker at the 7th International Workshop on Applications of Natural Language to Information Systems, NLDB'02, Stockholm, Sweden, June 2002.

14 Invited Speaker at the 1st International Semantic Web Conference (ISWC), Sardinia, Italy, June 2002.

15 Invited Speaker at the XML - Web Service One Conference, Track Web Services - The Future, San Jose, California, June 2002.

16 Invited Speaker at the 5th International Conference on Business Information Systems (BIS 2002), Poznan, Poland, April 2002.

17 Invited Speaker at the CollECTeR (Europe) Conference on Electronic Commerce, Toulouse, France, April 2002.

18 Invited Speaker at the Second Annual Diffuse Conference, Brussel, Belgium, February 2002.

19 Invited Speaker at Oracle, Ecommerce department, Redwood Shors, California, October 2001.

20 Invited Speaker at the 5th International Tagung WirtschaftsInformatik (WI2001), Augsburg, Germany, September 2001.

21 Penalist on the Semantic Web during the 9th International Conference on Cooperative Information Systems (CoopIS 2001), Trento, Italy, September 2001.

22 Penalist on the Semantic Web during the Semantic Web Working Symposium (SWWS-1), Stanford, California, July 2001.

23 Keynote Speaker of the 2nd International Workshop on Data Intensive Systems (UIDIS 2001), Zurich, Switzerland, May 31- June 1, 2001.

24 Penalist on the Semantic Web during the 10th World Wide Web Conference (WWW2001), Hong Kong, May 2001.

25 Keynote Speaker of the 9th IFIP 2.6 Working Conference on Database Semantics (DS-9), Hong Kong, April 2001.

26 Invited Speaker of the 2nd International Scientific and Practical Conference on Programming UkrPROG'2000, Ukraine, Kiev, May 2000.

27 Invited Speaker of the workshop Modellierung 2000, organiest by the GI Fachgruppen 0.0.1, 1.5.1, 2.1.6, 2.1.9, 2.5.2, 5.1.1, and 5.2.1, Koblenz-Laundau, April 2000.

28 Y. Sure. OntoEdit: Guiding Ontology Development by Methodology and Inferencing. International Conference on Ontologies, Databases and Applications of SEmantics ODBASE 2002, October 28 - November 1, 2002, University of California, Irvine, USA.

29 Y. Sure and V. Iosif. First Results of a Semantic Web Technologies Evaluation. Presented at: Common Industry Program at the federated event co-locating the three international conferences: DOA'02: Distributed Objects and Applications; ODBASE'02: Ontologies, Databases and Applied Semantics; CoopIS'02: Cooperative Information Systems DOA/ODBASE/CoopIS'02, October 28 -

November 1, 2002, University of California, Irvine, USA.

30 Y. Sure. OntoEdit: Collaborative Ontology Engineering for the Semantic Web. First International Semantic Web Conference 2002 (ISWC 2002), June 9-12 2002, Sardinia, Italia.

31 Y. Sure. Kollaborative Ontologie-Entwicklung. Invited talk: 11. Workshop der AG "Anwendungsszenario Fertigung", im Rahmen des DFG-SPP 1083 "Intelligente Softwareagenten und betreibswirtschaftliche Anwendungsszenarien", Frankfurt, Institut für Wirtschaftsinformatik (IWI), 15.05.2002.

32 Th. Lau und Y. Sure. Einführung eines Ontologie-basierten Skillmanagement Systems bei Swiss Life. Modellierung 2002, Modellierung in der Praxis - Modellierung für die Praxis, Tutzing, Deutschland, 25.-27. März 2002.

33 Y. Sure. A Methodology and Tools for Application-driven Ontology Development. Invited talk: Network Kick off Workshop of the COST Framework, Bremen, Germany, 1st-3rd November 2001.

34 Y. Sure. A Tool-supported Methodology for Ontology-based Knowledge Management. Annual Meeting of the FG Wissensmanagement (FGWM) 2001, Dormund, Germany, 10th-11th October 2001.

35 Y. Sure. State of the art report on Ontoknowledge. Annual Meeting of the FG Wissensmanagement (FGWM) 2001, Dormund, Germany, 10th-11th October 2001.

36 Y. Sure. Support for the roll-out of Knowledge Management Applications. OntoWeb Kick-off Meeting, Iraklion, Greece, 15th June 2001.

37 Y. Sure. Wissensvernetzung. Zukunftsforum FKT, Bühler Höhe, Deutschland, 09.-10. März .2001

38 Y. Sure. Ontoknowledge: Content-driven Knowledge Management through Evolving Ontologies. KM Europe 2000, Brussels, Belgium, 20th September 2000.

39 “Knowledge Management", master course at University of Karlsruhe, summer semesters, regulary, targeted audience: master students.

40 “Information and Knowledge Management", in collaboration with the Faculty for Informatics of University of Karlsruhe, master course at University of Karlsruhe, summer semesters, regularly, targeted audience: master students.

41 “Intelligent Systems for the WWW", master course at University of Karlsruhe, summer semesters, regularly, targeted audience: master students.

42 “Knowledge Management Curriculum”, University of Karlsruhe in collaboration with University of Chemnitz, advanced training, summer semesters, regularly,

targeted audience: employed persons that attend further education for knowledge management.

43 "Ontology Engineering for Knowledge Management", seminar at University of Karlsruhe, summer semester 2000, targeted audience: master students.

44 "Semantic Web", seminar at University of Karlsruhe, winter semester 2000/summer semester 2001, targeted audience: master students.

45 "Semantic Web Mining", seminar at University of Karlsruhe, summer semester 2001, targeted audience: master students.

46 "Knowledge Portals", seminar at University of Karlsruhe, winter semesters 2001-02/2002-03, targeted audience: master students.

47 "E-Business and Intelligent Web", seminar at University of Karlsruhe, winter semesters 2001-02/2002-03, targeted audience: master students.

48 "Cooperative Ontology and Metadata Engineering", seminar at University of Karlsruhe, summer semester 2002 targeted audience: master students.

49 "Web Services", seminar at University of Karlsruhe, winter semester 2002-03, targeted audience: master students.

50 “Semantic Technologies for Enterprises”, University of Karlsruhe in collaboration with Ontoprise Gmbh, industrial seminar, already held for over 5 times, targeted audience: companies.

7.4 Tool descriptions

7.4.1 CORPORUM - OntoExtract

The OntoExtract software shows the following features:

7.4.1.1 OntoExtract

Application level: OntoExtract main component

- webcrawling agent component implemented,
- filter for HTML, ASCII, PDF & evt. MS-Office integrated
- email notification system implemented
- parameters available: set domain URL to analyse, recursion depth (within domain), summary length (#sentences), size of concept vector, treshold on relation strength,

comment: the OntoExtract application has the capability to operate autonomously on substantial websites. Throughput is max. 60.000 documents/day. Main bottleneck is network connection speed and upload to external repositories (slow protocols).

Application level: linguistic kernel component:

- generates light-weight ontologies from natural language documents,
- performs automatic summary extraction
- extracts most relevant concepts and keywords
- identifies semantically related terms within documents
- exports in RDF(S)/DAML+OIL format
- parameters available: set base URLs, summary length (#sentences), size of concept vector, treshold on relation strength,

comment: some of the parameters are propagated from the kernel component to the application level. We have choosen to report them on both levels.

Functionality: linguistic kernel:

- analyses natural language document contents, using (somewhat "deeper") parsing techniques.
- Performs morphologic analysis and POS-tagging,
- proceeds with semantic annotation of relations.
- identifies and separates 'nouns' from 'proper nouns'

comment: the linguistic kernel is robust and relatively fast (600 ms for a 6 Kb ASCII text, equivalent to a 3 to 4 pages Word document. This is approx. the average size of a web document with text content.)

General shortcomings and future development:

- speed: improve speed of analysis, export and ontology uploading process
- correctness: improve NL parsing and POS tagging. This is needed in order to improve the quality of the semantic relations within the semantic network which is used for light-weight ontology building,
- exports: only supporting RDF(S) and DAML+OIL is not the final answer.
- an issue arose during Case Study development, and this was the issue of using external (background) sources for ontology development. Earlier versions of OntoExtract performed exactly this task, making use of a background repository in Sesame. However, the level of detailedness of the results was not regarded as applicable by the users.

For example, only few users where interested in selecting all references to “concept: car, property:colour, property_value: red”. Instead there was interest in just searching for “red cars”. I.e. there was too much information and a too complex representation. After this conclusion was drawn, we cut off the background analysis component.

However, we feel that there is a need for different kind of background knowledge probably more on the “synonym/antonym” level.

7.4.1.2 OntoWrapper

The OntoWrapper toolset is toolbox working in close cooperation with OntoExtract. It is a tool for users that know their domain and want to collect/screen for specific items such as name/phonenumber combinations. It has the following properties:

- user-specific repository: each user has its own area within the rule-database to define his/her business rules in,
- business rules can be applied in much the same way as OntoExtract, simply by defining a domain URL and a recursion depth together with the appropriate rule set,
- business rules are defined using templates and variables containing RDF(S) statements and the variables to search for in the target document.
- results of analysis are automatically uploaded to the Sesame database.

Comment: this tool is mainly meant for analysis of items/properties/entities mentioned in 'regular' webpages (directories, automatically generated pages like eBay etc.).

7.4.1.3 TableAnalyser

TableAnalyser is a piece of software developed to Beta-stage within the project. Development discontinued after Swiss Life left the project in favour of the original development plan.

The Swiss Life case study proposed a scenario where a different type of software could be utilised. Besides of OntoExtract there was a clear need for a piece of software which could take care of non-natural language information, mostly table based information. This is a scenario that was not foreseen within the consortium when defining the project before submission of the workplans. CognIT has decided to put in effort here in order to support and facilitate Swiss Life with some Use Case-driven software. However, when the Use Case was discontinued (Swiss Life leaving the consortium), there was no other Use Case requiring similar software. Therefore development of TableAnalyser has stopped at this point.

Functionality TableAnalyser component

- the TableAnalyser takes as input a 'table-ised' page. The tables are recursively analysed (often tables are used within tables, etc.) and represented as recursive objects. Each cell in the table is represented by an object with a value which can be a table in its own right.
- the user can select rows/columns to collaps and to combine.
- the system analysis all pages that are requested with the “table-rules” defined by the user
- after analyis the results are filled out in a rule-template (cf. OntoWrapper) and submitted to the repository (Sesame) in RDF(S)

This software is available in a (nearly untested) Beta version as part of the results of the OTK project. However, since the software was not build as part of the overall project plan, but rather as a support for a specific consortium partner who discontinued cooperation within the project, there is no reason for further development of this piece of software.

7.4.2 OntoEdit

OntoEdit is an ontology engineering environment that is rather unique in its kind as it combines methodology-based ontology development with capabilities for collaboration and inferencing. OntoEdit is a tool which enables inspecting, browsing, codifying and modifying ontologies and supports in this way the ontology development and maintenance task. Modeling ontologies using OntoEdit means modeling at a conceptual level, viz. (i) as much as possible independent of a concrete representation language, (ii) using GUI's representing views on conceptual structures (concepts, concept hierarchy, relations, axioms) rather than codifying conceptual structures in ASCII. The conceptual model of an ontology is internally stored using a powerful ontology model, which can be mapped onto different, concrete representation languages. It is based on a modular and flexible plugin framework that allows for extending the core functionalities very easily. In the following we will describe the plugins that were developed in the On-To-Knowledge project.

SesameClientPlugin: This plugin is basic infrastructure for the On-To-Knowledge project as it realizes the connection from OntoEdit to the Sesame repository from AIdministrator. The plugin allows for uploading and downloading of ontologies between OntoEdit and Sesame. E.g. one can download an ontology (including data statements) from Sesame, modify it in OntoEdit, and upload it again into the Sesame repository. The plugin is able to handle multiple sessions on different Sesame repositories. The current version includes also features on top of the Ontology Middleware Module (OMM) to enable versioning facilities for ontologies in a Sesame repository. Three protocols can be used for the communication between OntoEdit and Sesame: HTTP, RMI and SOAP.

Mind2Onto: This tool connects the commercially available mind mapping tool MindManager 2002 Business Edition (http://www.mindjet.com) with OntoEdit. Mind Maping is a brain storming technique that is well suited for the early stages of ontology development. The core functionality of Mind2Onto consists of support for the transformation process of mind maps into ontologies, including the distinction of concepts and relations, the creation of unique identifiers for concepts and relations (by using namespaces) and an XML based import and export facility that connects the MindManager with OntoEdit.

OntoKick: This plugin supports the Kick-off phase of ontology engineering. Users can capture requirements for an ontology as proposed in the methodology. The main functionality is support for capturing Competency Questions (CQ) from domain experts. Out of these CQs the most relevant concepts and relations for a domain can be extracted. The plugin stores the linkage between concepts and relations and their related CQs, thereby offering traceability of the origins of concepts and relations during later stages of the ontology engineering process.

OntoFiller: This plugin allows users to easily "fill in" external representations for concepts. The main purpose is to support users during the translation of an ontology into different languages (e.g. German, English, French and Italian - as needed in the SwissLife case study). The basic functionality consists of the generation of language-specific views on the ontology. Users can easily identify not yet translated concepts (or relations) and fill in external representations for various languages. Additionally, this plugin contains a translation support on top of the free German-English dictionary LEO (http://www.leo.org). A user can request "translation hints" from this online dictionary for concepts and relations. They consist of translations for a chosen concept or relation from German to English (and vice versa). A user then might choose from the received hints the appropriate translation for his purpose. As a side effect this plugin allows for viewing and editing the namespaces used for concepts and relations.

OntoCleanPlugin: This proof-of-concept implementation allows for formal evaluation of ontologies according to the well-known OntoClean methodology.

7.4.3 QuizRDF

QuizRDF is a search facility for the Semantic Web. This patented technology offers the next generation of intelligent search based on Resource Description Framework (RDF) technology. RDF is a language that allows WWW-based information to be presented in a format that can be processed by machines. Today's search engines are based on a free text search method. The user enters a search query consisting of one or more words and the search engine returns details of pages that contain those words. A typical search will return many hundreds of pages and many of these will be of little use to the user. Words can, of course, have many different meanings in different contexts, and computers have no way of distinguishing exactly which meaning the user or the page author had intended. The search engine exploits RDF data to provide users with the means to express exactly what they are looking for. After entering a textual query, they are presented with a set of concepts from which they can select in order to significantly narrow down their search results to a set that is relevant to them. It can either be used in conjunction with an RDF repository such as Sesame or it can be used in it's own right to extract RDF from source pages. QuizRDF is a fully developed prototype and has been used in OntoKnowledge case studies. We are currently seeking partners to help us further develop and trial the software.

7.4.4 OntoShare

OntoShare is a system designed to facilitate and encourage the sharing of information between communities of practice within, or perhaps across, organisations and to encourage people - who may not previously have known of each other's existence in a large organisation - to make contact where there are mutual concerns or interests. When a user finds information of sufficient interest to be shared with their community of practice, they can add it to the system's store. The system examines the document and determines to which of a set of "concepts" it should be added. These concepts comprise a hierarchy of subject areas describing the domain in which the community operates - in other words, an "ontology". Ontologies are a key aspect of the Semantic Web in that they provide a framework for the classification and description of data resources and common understanding for a domain. As well as classifying the document, the sharing system notifies other users of its existence. It only informs users to whom it believes the document will be relevant - based upon their membership profile to the different concepts in the ontology. This helps to prevent information overload - an important issue in knowledge management. OntoShare is able to export its data store as RDF. This means that meta-information about the documents stored (including summaries, keywords, concepts, etc.) can be used by other applications such as Sesame or other knowledge management tools. OntoShare is a fully developed prototype and has been used in OntoKnowledge case studies. We are currently seeking partners to help us further develop and trial the software

7.4.5 Sesame

General idea: Sesame an architecture for efficient storage and expressive querying of large quantities of RDF data. Sesame provides facilities for

- uploading RDF data to a permanent-storage repository,

- evaluating RQL and RDQL queries over the data in such a repository
- simple administrative operations on such a repository

Sesame is compliant with the full RDF/RDF Schema model theory. The architecture provides a separate abstraction layer (SAIL) for hiding the detail of specific storage technology. The archictecture also provides handlers for multiple protocols that can be used to address Sesame.

Specific Implementation: An open-source Java (v1.3) implementation, providing handlers for SOAP, HTTP and RMI, and providing support for storage in PostgreSQL, MySQL, Oracle 9i and main memory. For further details, see the fact-sheet at http://sesame.aidministrator.nl.

Sesame is made available under the GNU Lesser General Public License (LGPL). As a consequence of this, any further developments to Sesame will also have to be made available under this license.

7.4.6 Spectacle

The spectacle presentation and visualisation software comprises both the Spectacle ClusterMap and the Spectacle Website Generator.

7.4.6.1 Spectacle ClusterMap

General idea: ClusterMaps visualise a set of instantiated classes by means of a graph which is organised on the basis of the overlapping instances between these classes. Algorithms compute a graphical layout where the geometrical proximity in the diagram reflects the semantic proximity between the classes. The resulting visualisation represents the semantic relations between the classes.

The visualisation can be configured in a number of different ways, such as (among others) use of color-coding, size of displayed objects, usage of different icons.

Different types of clustermaps can be used (among other purposes) for the analysis of datasets, comparing different datasets, for displaying the results of querying datasets, and for navigating within datasets.

Specific Implementation: A set of software components and documented API's (Java v1.4) for

1. constructing a graphical representation of a collection of instantiated classes
2. display of and interaction with this graphical representation in a graphical user interface
3. exporting this graphical representation in external PNG format and as HTML image map
4. importing the instantiated classes from an external source
    a. either in RDF Schema format using the Sesame Client library
    b. or in a specified XML format

and storing these instantiated class for future reuse in the XML format.

7.4.6.2 Spectacle Website Generator

The Spectacle Website Generator takes as input an RDF Schema taxonomy stored in Spectacle, and produces a website whose navigation structure is based on this taxonomy. Spectacle differs crucially from other tools in that it does *not* take the taxonomy as the navigation structure, but that more sophisticated strategies are used to *generate* the navigation structure from this taxonomy, via the following steps:

1. identify relevant information entity types in the data sources and the data to display for each entity. For each entity type:
2. create a mapping of terms, relations, properties and property values from the domain ontology to taxonomical terms
3. Decide on how to render the individual entities in the presentation
4. Decide on the navigation paths through the ontology that are relevant to the entire entity set, based on the properties from step 2.
5. Decide on which information to render in each step of the navigation sequence