Ontologies and their Role in Knowledge Management and E-Business Modelling

Hans Akkermans
Free University Amsterdam VUA, The Netherlands,contact: elly@cs.vu.nl

Abstract: Ontologies are reference conceptual models that formally describe the consensus about a domain and that are both human-understandable and machine processable. Ontologies are a key technology to realising the next, smarter, generation of the World Wide Web, known as the Semantic Web. We give an overview of recent developments, issues, and experiences in Semantic Web research, and especially discuss the role of ontologies in innovative intelligent e-applications. This paper discusses as a particular example the On-To-Knowledge project for ontology-based knowledge management. It aims to speed up knowledge management, dealing with large numbers of heterogeneous, distributed, and semi-structured documents typically found in large company intranets and the World Wide Web, by: (1) a toolset for semantic information processing and user access; (2) OIL, an ontology-based inference layer on top of the World Wide Web; (3) validation by industrial case studies in knowledge management.

1 INTRODUCTION

The World Wide Web (WWW) has drastically changed the availability of electronically available information. Currently, there are around one billion documents in the WWW, which are used by more than 300 million users internationally. And that number is growing fast. However, this success and exponential growth makes it increasingly difficult to find, to access, to present, and to maintain the information required by a wide variety of users. The competitiveness of many companies depends heavily on how they exploit their corporate knowledge and memory. Most information in modern electronic media is mixed media and rather weakly structured. This is not

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35621-1_43

K. Kosanke et al. (eds.), *Enterprise Inter- and Intra-Organizational Integration*

only true of the Internet but also of large company intranets. But as volumes of information continue to increase rapidly, the task of turning them into useful knowledge has become a major problem. Tim Berners-Lee envisioned a *Semantic Web* (cf. Berners-Lee et al., 2001) that provides automated information access based on machine-processable semantics of data and heuristics that use these meta-data. The explicit representation of the semantics of data, accompanied with domain theories (i.e., ontologies), will enable a web with various specialised smart information services that will become as necessary to us as access to electric power.

Ontologies (cf. Staab et al., 2001, Fensel, 2001) are a key enabling technology for the semantic web. They aim to interweave human understanding of symbols with their machine processability. Ontologies were developed in artificial intelligence to facilitate knowledge sharing and re-use. Since the early nineties, ontologies have become a popular research topic. They have been studied by several artificial intelligence research communities, including knowledge engineering, natural language processing and knowledge representation. More recently, the concept of ontology is also gaining tremendous ground in fields, such as intelligent information integration, cooperative information systems, information retrieval, electronic commerce, and knowledge management. The reason ontologies are becoming so popular is largely due to the fact that they cater for an important general need: *a shared and common understanding of a domain that can be communicated between people and application systems.*

Other applications and case studies on the use of ontologies in e-business modelling have been published elsewhere (Akkermans, 2001, Gordijn and Akkermans, 2001, Schulten et al. 2001).

2 TOOL ENVIRONMENT FOR ONTOLOGY-BASED KNOWLEDGE MANAGEMENT

A major objective of the On-To-Knowledge 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. Most of the tools presented here in Fig. 1 are described below.

RDF*ferret* combines full text searching with RDF querying. It can be used like a conventional Internet search engine by entering a set of search terms or a natural language query and produces a list of links to relevant Web pages in the usual way. However, RDF*ferret*'s indexing and retrieval technique is also designed to use domain knowledge that is made available in the form of ontologies specified as RDF Schemas. The information items processed by RDF*ferret* are RDF resources, which may be Web pages or parts thereof and such pages or segments are effectively ontological instances. During indexing RDF*ferret* assigns content descriptors to RDF resources: terms (words and phrases) that RDF*ferret* obtains from a full text analysis of the resource content and from processing all literal values that are directly related by a property. They also retain structural information about the ontology. In RDF*ferret* the user can select from a list of all the resource types stored in the index. When searching by selecting a resource type, RDF*ferret* adjusts its result list to show only resources of the selected type. The user is also presented with a search and navigation area. The search area shows the attributes of the selected resource type. For each attribute the user can input a search criterion. RDF*ferret* combines the search criteria entered and matches the resulting query against its ontology-based index. In addition, resource types (ontological classes) related by some property to the currently selected type are displayed as hyperlinks. Clicking on such a type then selects that type and in turn displays those types that are related to it. Thus, the user can browse the ontology in a natural and intuitive way. Fig. 2 shows a typical initial query by a user. The user has entered a query for information about an employee called "George Miller". The search engine has returned a ranked list of 73 documents mentioning the terms "George" and/or "Miller". At the top of the screenshot can be seen a drop-down list containing the selection "any...". When returning the 73 results documents, RDF*ferret* has also compiled a list of the classes to which each document belongs. This class list is made available to the user via the drop-down list.

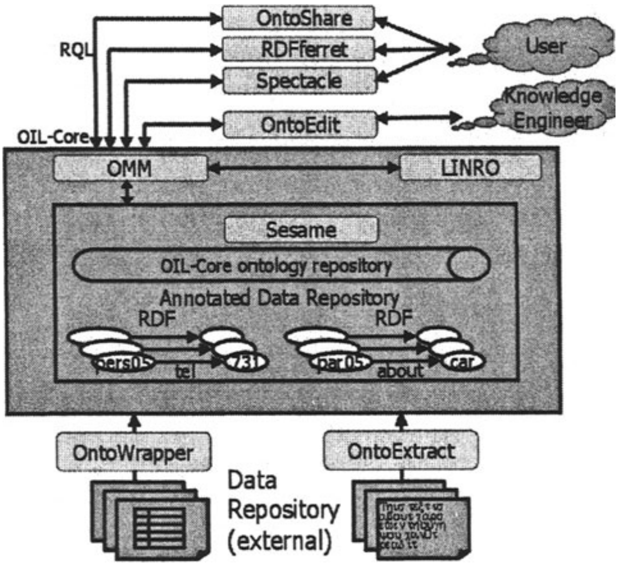
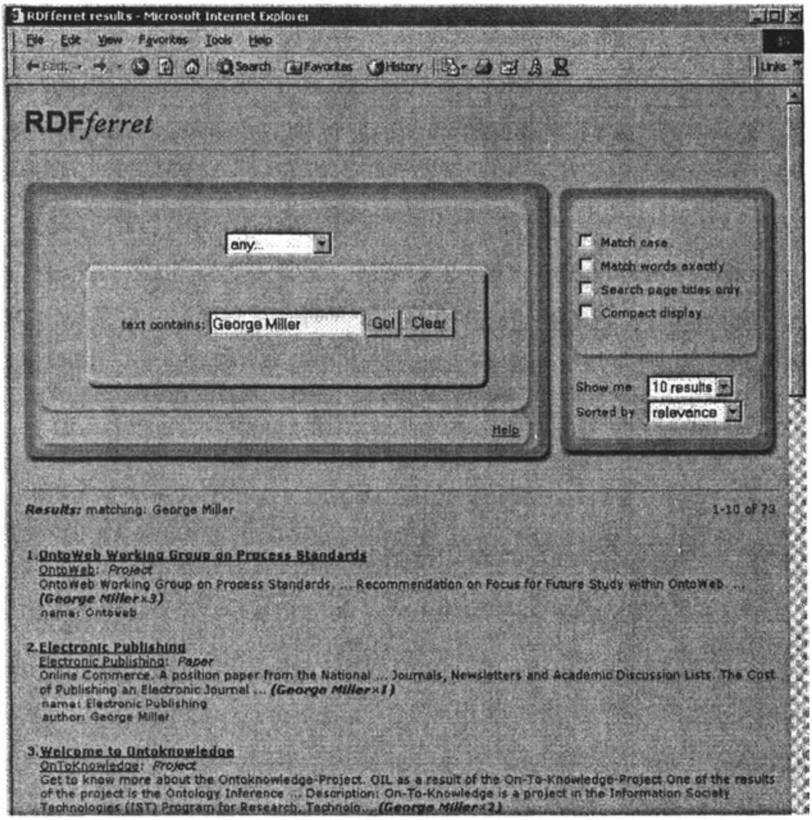


Figure 1: The technical architecture of On-To-Knowledge.

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. In addition, the ontology helps users to become familiar with new domains. It provides a sharable structure for the knowledge base, and a common language for communication between user groups.



Spectacle organises the presentation of information. This presentation is ontology driven. Ontological information, such as classes or specific attributes of information, is used to generate domain exploration contexts for users. The context is related to certain tasks, such as finding information or buying products. The context consists of three modules: (1) Content: specific content needed to perform a task; (2) Navigation: suitable navigation disclosing the information; (3) Design: applicable design displaying the selected content. The modules are independent. Spectacle consists of the following parts:

- Spectacle server, which handles all interaction between users and exploration contexts;
- Libraries for creating large scale exploration contexts in this server;
- Graphical user interface for building small-scale exploration contexts.

OntoEdit (Sure et al., 2002) 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.

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 (OMAPI) 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 (such as Status, Last-Updated-By, Responsible, Comments, etc.) for whole ontologies, as well as for separate concepts and properties.
- Access via several protocols: HTTP, RMI, EJB, CORBA, and SOAP.

Sesame (Broekstra et al, to appear) is a system that allows persistent storage of RDF data and schema information and subsequent online querying of that information. Sesame has been implemented in Java, which makes it portable to almost any platform. It also abstracts from the actual repository used by means of a standardised API. This API makes Sesame portable to any repository (DBMS or otherwise) that is able to store RDF triples. At the same time, this API enables swift addition of new modules that operate on RDF and RDF Schema data. One of the most prominent modules of Sesame is its query engine. It 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. The streaming approach used in Sesame (data is processed as soon as available) makes for a minimal memory footprint. This streaming approach also makes it possible for Sesame to scale to huge amounts of data. Sesame can scale from devices as small as palm-top computers to powerful enterprise servers.

The CORPORUM toolset *(OntoExtract and OntoWrapper)* (Engels & Bremdal, 2000) has two related tasks: interpretation of *natural language* texts and ontology extraction of *specific information* from free text. The latter task requires a user who defines business rules for extracting information from tables, (phone) directories, home pages, etc. The former task 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.). CORPORUM outputs a variety of (symbolic) knowledge representations, including semantic (network) structures and visualisations thereof, lightweight ontologies, text summaries, automatically generated thesauri (related words/concepts), etc. Extracted information is represented in RDF(S)/DAML+OIL, augmented with Dublin Core Meta Data wherever possible, and submitted to the Sesame Data Repository. CORPORUM does not incorporate background knowledge itself, but relies on any knowledge available in the Sesame repository.

3 *OIL*: INFERENCE LAYER FOR THE SEMANTIC WORLD WIDE WEB

The tools discussed in section 2 all exploit ontologies as their common operating ground. All of this requires the existence of a language to express such ontologies. Some basic requirements for such a language are:

- Sufficient expressivity for the applications and tasks (sketched elsewhere in this paper);
- Sufficiently formalised to allow machine processing;
- Integrated with existing Web technologies and standards.

Although much work has been done on ontology languages in the AI community (see e.g. (Corcho & Gomez Perez, 2000) for a recent overview), it is particularly the third requirement that motivated us to design a new language (baptised OIL) for our purposes. In this section, we will briefly describe the constructions in the OIL language, and then discuss its most important features and design decisions.

Combining Description Logics with Frame Languages. The OIL language (Harmelen & Horrocks, 2000, Fensel et al., 2000) is designed to combine frame-like modelling 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 serialisations (as well as a formally specified "human readable" form, see OIL, http://). Classes (concepts) are described by frames, which consist of a list of super-classes and a list of slot-filler pairs. A slot corresponds to a role in a DL, and a slot-filler pair corresponds to either a universal value restriction or an existential quantification. OIL extends this basic frame syntax so that it can capture the full power of an expressive description logic. These extensions include:

- Arbitrary Boolean combinations of classes (called class expressions) can be formed, and used anywhere a class name can be used. In particular, class expressions can be used as slot fillers, whereas in typical frame languages slot fillers are restricted to being class (or individual) names.
- A slot-filler pair (called a slot constraint) can itself be treated as a class: it can be used anywhere that a class name can be used, and can be combined with other classes in class expressions.
- Class definitions (frames) have an (optional) additional field that specifies whether the class definition is primitive (a subsumption axiom) or non-primitive (an equivalence axiom). The default is primitive.
- Different types of slot constraints are provided for universal value restrictions, existential quantification, various cardinality constraints.
- Global slot definitions allow for the specification of superslots (subsuming slots) and of properties such as transitivity and symmetry.
- Unlike frame languages, no restriction exists on the ordering of class and slot definitions, so classes and slots can be used before they are defined.
- OIL also provides axioms for asserting disjointness, equivalence and coverings with respect to class expressions.

Many of these points are standard for a description logic, but are novel for a frame language.

Web Interface. 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 subproperties), and to define domain and range constraints on properties. Instances of such classes and properties are defined in RDF. OIL has been designed to be a superset of the constructions in RDF Schema: all valid RDF Schema expressions are also valid OIL expressions. Furthermore, 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 subdocument is guaranteed to be sound (but of course incomplete) with respect to the interpretation of the full OIL document. This guarantees that any RDF Schema agent can correctly process arbitrary OIL documents, and still correctly capture some of the intended meaning. The full details of how this has been achieved, and the trade-offs involved in this can be found in (Broekstra, *et al.*, 2000).

Layering. For many of the applications from section 1, 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 sublanguage 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. The standard language, is called "Standard OIL", and when extended with the ability to assert that individuals and tuples are, respectively, instances of classes and slots), is called "Instance OIL". Finally, "Heavy OIL" is the name given to a further layer that will include as yet unspecified language extensions. This layering is depicted in Fig. 3.

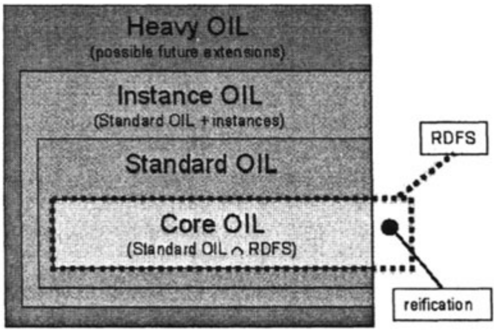


Figure 3: The layered language model of OIL.

Current status. Meanwhile, OIL has been adopted by a joined EU/US initiative that developed a language called DAML+OIL (http://), which has now been submitted to the Web Ontology Group of the W3C (http://), the standardisation committee of the WWW. We can soon expect a recommendation for a web ontology language; it features many of the elements on which OIL is based.

Future developments: OWL. In November 2001, the W3C started a Working Group for defining a Web Ontology language. This WG is chartered to take DAML+OIL as its starting point. Over 40 of the W3C members from academia and industry are currently participating in this effort. It is most likely that such a Web Ontology language will range in power somewhere between the rather simple RDF Schema and the rather rich Standard OIL language. Other efforts are underway to define extensions for this web ontology language - that has been named OWL - such as an ontology-query language, or an extension with rules (which would allow for example role chaining, as done in Horn logic).

4 BUSINESS APPLICATIONS IN SEMANTIC INFORMATION ACCESS

Accounting Information Search. Swiss Life carried out two case studies to evaluate the developed Semantic Web tools and methods. 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. With the help of the ontology extraction tool OntoExtract, an ontology was automatically learned from the document. The ontology consists of 1,500 concepts linked by 47,000 weighted semantic associations. It supports users in reformulating their initial queries when the results fall short of expectations, by offering terms from the ontology that are strongly associated with (one of) the query terms used in the initial query. An evaluation of user behaviour showed that 70% of the queries involved a reformulation step. On average, 1.5 refinements were made. Thus, although the ontology is structurally quite simple, it greatly improves search results. Another advantage to using a simple ontology is that it requires no manual effort to build.

Skills Management. Swiss Life's second case study is a skills management application that uses manually constructed ontologies about skills, job functions, and education. These consist of 800 concepts with several attributes, arranged into a hierarchy of specialisations. There are also semantic associations between these concepts. The skills management system makes it easy for employees to create in a personal home page on the company's intranet that includes information about personal skills, job functions, and education. The ontology allows a comparison of skills descriptions among employees, and ensures the use of uniform terminology in skills descriptions and queries for employees with certain skills. Moreover, the ontology can automatically extend queries with more general, more specialised, or semantically associated concepts. This enables controlled extension of search results, where necessary.

Exchanging knowledge in a virtual Organization. The case study done by EnerSearch AB focuses on satisfying the information dissemination needs of a virtual organisation. The goal of the case study is to improve knowledge transfer between EnerSearch's in-house researchers and outside specialists via the existing web site. The study also aims to help the partners from shareholding companies to obtain up-to-date information about research and development results. The main problem with the current web site is that its search engine supports free text searches rather than content-based information retrieval, which makes it fairly difficult to find information on certain topics. To remedy this, the entire web site was annotated by concepts from an ontology developed using semi-automatic extraction from documents on the EnerSearch's current web site. The RDF*ferret* search engine is used to extend free text searches to searches of annotations. Alternatively, the Spectacle tool enables users to obtain search results arranged into topic hierarchies, which can then be browsed. This offers users a more explorative route to finding the information they need (see Fig. 4). Three groups with different interests and needs are involved in the evaluation: (1) researchers from different fields, (2) specialists from the shareholders organisation and (3) outsiders from different fields.

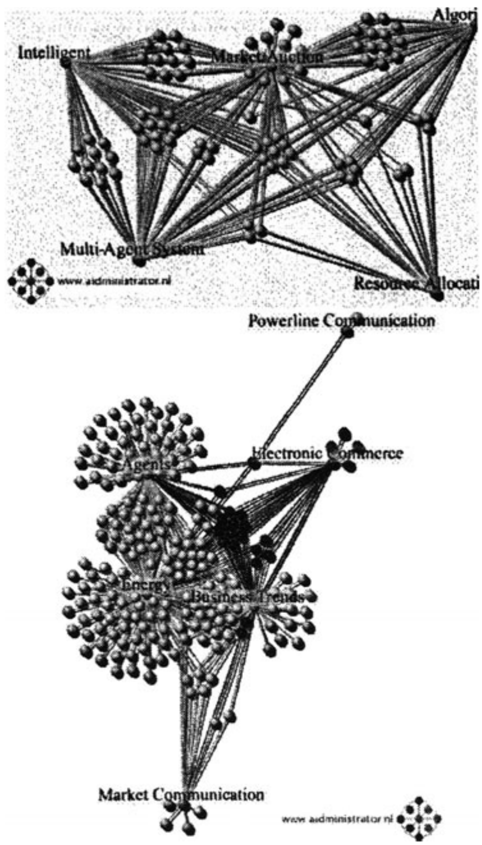


Figure 4: Automatically generated semantic structure maps of the EnerSearch website.

5 CONCLUSION

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. *On-To-Knowledge* 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.

We also encountered a number of shortcomings in our current approach. Building ontologies that are a pre-requisite for - and result of - the common understanding of large user groups is no trivial task. A model or "protocol" that maintains the process of *evolving* ontologies is the real challenge for making the *semantic* web a reality. Most work on ontologies views them in terms of an isolated theory containing a potentially large number of concepts, relationships, and rules. In practice, ontologies must be structured as interwoven *networks* that make it possible to deal with heterogeneous needs in the communication processes that they are supposed to mediate. Moreover, these ontologies change over time because the processes they mediate are based on consensual representation of meaning. It is the network of ontologies and their dynamic nature that make future research progress necessary. Actual research challenges on ontologies are what glue keeps ontology networks together in space and time. Instead of a central, top-down process, we require a distributed process of emerging and aligned ontologies. Most existing technology focuses on building ontologies as graphs based on concepts and relationships. Our current understanding is insufficient when it comes to proper methodological and tool support for building up networks, where the nodes represent small and specialised ontologies. This is especially true of the noisy and dynamically changing environment that the web is and will continue to be.

6 ACKNOWLEDGEMENTS

This paper and the research work it describes is based on contributions from many people, in particular Dieter Fensel, Frank van Harmelen, Peter Mika, Michel Klein (Free University Amsterdam VUA), Jeen Broekstra, Arjohn Kampman, Jos van der Meer (Administrator, The Netherlands), York Sure, Rudi Studer (University of Karlsruhe, Germany), John Davies, Alistair Duke (BT, Ipswich, UK), Robert Engels (CognIT, Oslo, Norway), Victor Iosif (Enersearch AB, Malmö, Sweden), Atanas Kiryakov (OntoText, Sofia, Bulgaria), Thorsten Lau, Ulrich Reimer (Swiss Life, Zürich, Switzerland), and Ian Horrocks (University of Manchester, UK). It has been partially supported by the European Commission through the EU-IST project On-To-Knowledge (IST-1999-10132).

7 REFERENCES

Akkermans, J.M. (2001), *Intelligent E-Business – From Technology to Value*, IEEE Intelligent Systems Vol. 16, No. 4, pages 8-10. Special issue on Intelligent E-Business. Also available from http://computer.org/intelligent.

Berners-Lee, T. Hendler, J. Lassila, O. (2001), *The Semantic Web*, Scientific American, May.

Broekstra, J. Klein, M. Decker, S. Fensel, D. van Harmelen, F. Horrocks. I. (2001), *Enabling knowledge representation on the web by extending RDF schema.* In Proceedings of the Tenth International World Wide Web Conference (WWW10), Hong Kong, May.

Broekstra, J. Kampman, A. van Harmelen, F. (to appear 2002), *Sesame: An Architecture for Storing and Querying RDF Data and Schema Information.* In Fensel, D. Hendler, J. Lieberman, H. Wahlster, W. (Eds.): Semantic Web Technology, MIT Press, Cambridge, MA, to appear.

Corcho, O. Gomez Perez, A. (2000), *A roadmap to ontology specification languages*. In R. Dieng and O. Corby (Eds.), Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW'00), volume 1937 of LNAI, 80–96. Springer-Verlag.

DAML+OIL, http://www.daml.org

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

Fensel, D. Horrocks, I. Van Harmelen, F. Decker, S. Erdmann, M. Klein, M. (2000), *OIL in a nutshell*. In R. Dieng and O. Corby (Eds.), Knowledge Engineering and Knowledge Management – Methods, Models and Tools, pages 1-16, Lecture Notes in Artificial Intelligence, LNAI 1937, Springer-Verlag,

Fensel, D. (2001), *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag.

Gordijn, J. Akkermans, J.M. (2001), *Designing and Evaluating E-Business Models*, IEEE Intelligent Systems Vol. 16, No. 4), pages 11-17. See http://computer.org/intelligent. Further related work: http://www.cs.vu.nl/~gordijn.

Harmelen, F. van, Horrocks, I. (2000), *Questions and answers about OIL*. IEEE Intelligent Systems, 15(6): 69–72.

OIL, http://www.ontoknowledge.org/oil/syntax/

Schulten, E. Akkermans, J.M. Botquin, G. Dörr, M. Guarino, N. Lopes, N. Sadeh, N. (2001), *The E-Commerce Product Classification Challenge*, IEEE Intelligent Systems Vol. 16, No. 4 (July-August), pages 86-89. (http://computer.org/intelligent).

Staab, S. Schnurr, H.-P. Studer, R. Sure, Y. (2001), *Knowledge Processes and Ontologies*, IEEE Intelligent Systems, Vol. 16, No. 1, pages 26-34.

Sure, Y. Erdmann, M. Angele, J. Staab, S. Studer R. Wenke, D. (2002), *OntoEdit: Collaborative Ontology Engineering for the Semantic Web*. In: Proceedings 1st International Semantic Web Conference 2002 (ISWC 2002), June, Sardinia, Italia.

W3C, http://www.w3c.org