The Semantic Web: Yet Another Hip?

Ying Ding, Dieter Fensel, Michel Klein, and Borys Omelayenko

Division of Mathmatics & Computer Science,
Vrije Universiteit Amsterdam,
De Boelelaan 1081a, 1081 HV Amsterdam, NL,

(order is currently alphabetically may be changed later)

Abstract. Currently computers are changing from single isolated devices into entry points to a worldwide network of information exchange and business transactions called the World Wide Web (WWW). For this reason, support in data, information, and knowledge exchange has become a key issue in current computer technology. The WWW has drastically changed the availability of electronically available information. However, this success and exponential growth has made it increasingly difficult to find, access, present, and maintain the information required by a wide variety of users. In response to this problem, many new research initiatives and commercial enterprises have been set up to enrich available information with machine processable semantics. Such support is essential for "bringing the web to its full potential". This semantic web will provide intelligent access to heterogeneous and distributed information, enabling software products (agents) to mediate between user needs and the information sources available. This paper summarizes ongoing research in the area of the semantic web especially focussing on ontology technology.

1 Introduction

The World Wide Web (WWW) has drastically changed the availability of electronically available information. The WWW currently contains some 500 million static documents, which are used by over 300 million users internationally. What is more, that number is growing astronomically. In 1990, the WWW began with a small number of documents as an in-house solution for approximately one thousand users at CERN. By 2002, the standardization committee for the WWW (called W3C) expects around a billion web users and an even higher number of available documents. However, this success and exponential growth has made it increasingly difficult to find, access, present, and maintain the information required by a wide variety of users because the information content is presented mainly in natural language. Thus, there is a wide gap between the information available for tools that try to address the problems above and the information maintained in human-readable form.

- **Searching and presenting information**: Even now, finding the right piece of information is often a nightmare. Searches are imprecise, often yielding matches to many thousands of pages. Moreover, users face the task of reading the documents retrieved in order to extract the information desired. A related problem is that the maintenance of web sources has become very difficult. The burden on users to maintain consistency is often overwhelming. This has resulted in a vast number of sites containing inconsistent and/or contradictory information.
- **Electronic commerce**: Automatization of electronic commerce is seriously hampered by the way information is currently presented. Shopping agents use wrappers and heuristics to extract product information from weakly structured textual information. However, development and maintenance costs are high and the services provided limited.

In response to this problem, many new research initiatives and commercial enterprises have been set up to enrich available information with machine processable semantics. Such support is essential for "bringing the web to its full potential". Tim Berners-Lee, Director of the World Wide Web Consortium, referred to the future of the current WWW as the "*semantic web*" -- an extended web of machine-readable information and automated services that extend far beyond current capabilities ([Berners-Lee et al., 2001], [Fensel & Musen, 2001]). The explicit representation of the semantics underlying data, programs, pages, and other web resources, will enable a knowledge-based web that provides a qualitatively new level of service. Automated services will improve in their capacity to assist humans in achieving their goals by "understanding" more of the content on the web and thus provide more accurate filtering, categorization, and search of information sources. This process will ultimately lead to an extremely knowledgeable system with various specialized reasoning services that will support us in nearly all aspects of our daily life -- making access to information as pervasive, and necessary, as access to electricity is today.

A key enabler for the semantic web is on-line ontological support for data, information and knowledge exchange. Given the exponential growth of the on-line information available, automatic processing becomes mandatory for keeping it managed and accessible. Used to describe the structure and semantics of information exchange, *Ontologies* will play a key role in areas, such as knowledge management, B2B e-commerce and other such burgeoning electronic enterprises (cf. [Fensel, 2001]).

The contents of the paper is organized as follows. In Section 2, we provide an introduction in the basic infrastructure of the semantic web. These are web-enabled languages allowing machine processable semantics of data and tools that can process this data. In Section 3, we discuss one technology in more detail. Actually we discuss Ontologies which could be characterized as the backbone of the semantic web because they provide formal and consensual domain models that can be used to interpret data both by humans and computers. Section 4 is about potential application areas: we have chosen as subjects of the discussion knowledge management and electronic commerce. Finally Section 5 provides conclusions and future trends.

2 Semantic Web Technology

In this section, we will sketch some languages and tools that provide the basic infrastructure of the semantic web.

2.1 Languages on the Web: Adding structure and meaning

An important requirement for machine-processable information is *structure* in data. There are several ways to structure data: databases, for example, use tables with attributes as structuring mechanism. The main structuring technique that is used on the Web is *document markup*. This means that certain sequences of characters in a document contain information indicating the role of the document's content. The markup describes the document's data layout and logical structure and makes the information self-describing, in a sense. The markup often takes the form of words between pointy brackets, called *tag*s—for example, `<title>` or `<h1>`.

A *markup language* is a specification of the markup-tags that may used, how they may combined, and what their meaning is. HTML is the most well-known and wide spread language on the web. It provides tags to specify abstract rendering information for text documents. For example, text between `<h1>` and `</h1>` is considered as a heading of the first level, while text between `<li>` and `</li>` is regarded as a list item. Browsers use this information to format web pages appropriately.

However, HTML only provides rendering information about the content, which is often not sufficient for the advanced automated services that are foreseen on the Semantic Web. Applications will need specialized markup that specifies the role of parts of the content that are relevant for them. For example, for a price comparison application on the Web, it is useful if all prices are annotated with a tag such as `<price>`. It is necessary to specify the *meaning* of the tags.

2.1.1 XML

The eXtensible Markup Language (XML) is a meta-language that meets the need to define application specific markup tags. XML is a mechanism for representing other languages in a standardized way. In other words, XML only provides a data format for structured documents, without specifying an actual vocabulary. A Document Type Definition (DTD) or an XML Schema might be used to specify this vocabulary and to define the allowed combinations of tags. A language that is defined in XML is called an XML application.

XML does not imply a specific interpretation of the data. On account of the tag's names, the meaning of the a piece of XML—for example `<person><name>`John`</name></person>`—seems sometimes obvious to human users, but it is not formally specified. The only legitimate interpretation is that XML code contains named entities with sub-entities and values; that is, every XML document forms an ordered, labeled tree. This generality is both XML's strength and its weakness. You can encode all kinds of data structures in an unambiguous syntax, but XML does not specify the data's use and semantics. The parties that use XML for their data exchange must agree beforehand on the vocabulary, its use, and its meaning.

2.1.2 RDF and RDF Schema

XML is a standard mechanism to *structure* data, the Resource

Description Framework (RDF) is a mechanism to *tell* something about data, i.e., to give meaning to it. As its name indicates, it is not a language but a model for representing data about “things on the Web.” This type of data about data is called *meta data*. The “things” are *resources* in RDF vocabulary. Resources are all things that have an identifier on the Web, ranging from specific, identifiable parts of a document to things like persons or companies.

Basically, the RDF data model is quite simple. It consist of statements about resources, encoded as object-attribute-value triples. The objects are resources and the values are resources or strings. For example, to state that “John” is the author of the article at a specific URL, one would use the triple: `http://www...`, author, “John”. Attributes, such as “author” in this example, are called *properties*.

RDF Schema is a simple type system for RDF. It provides a mechanism to define domain-specific properties and classes of resources to which you can apply those properties. The basic modeling primitives in RDF Schema are *class* definitions and *subclass-of* statements (which together allow the definition of class hierarchies), *property* definitions and *sub-property-of* statements (to build property hierarchies), *domain* and *range* statements (to restrict the possible combinations of properties and classes), and *type* statements (to declare a resource as an instance of a specific class). With these primitives you can build a schema for a specific domain.

Together, RDF and RDF Schema provides a simple knowledge representation mechanism for Web resources. However, RDF Schema is quite simple compared to full-fledged knowledge representation languages. It only has a few modeling primitives, and does not have exact, logical, semantics. To be able to specify the meaning of data more precisely, richer languages are necessary. Such languages will be discussed in the next section.

2.2 Tools

In the following, we will describe parsers and database technolgy of the semantic web (for more details see [Hjelm, 2001]).

2.2.1 RDF Parsers

RDF parsers provide the basic support in parsing different RDF serializations, accessing RDF triples via programming interfaces or queries, and providing basic operations with the triples.

The SiRPAC[1] RDF parser SiRPAC is a set of Java classes that can parse RDF/XML documents into the three-tuples of the corresponding RDF data model. The parser has evolved in several versions and has become a de-facto standard in Java-based RDF development.

The Profium[2] tool targets the same objective as SiRPAC and provides similar functionality. It is available as a Perl script in addition to its Java implementation.

The perllib[3] W3C library was born of a need to implement an RDF infrastructure at W3C. This is currently used for access control and annotations, but will be used for a more diverse group of applications as our needs evolve. The library is implemented in Perl and is now under the pre-release preparation.

There exists a number of other RDF-supporting tools and libraries, as well as a number of lists of those tools, see the Dave Beckett's RDF page[4] or W3C RDF page[5]. These parsers serve as a basic tool to build RDF repositories and databases.

ICS-FORTH[6] Validating RDF Parser is a tool for parsing RDF Statements and validating them against an RDF Schema. The parser analyses syntactically the statements of a given RDF/XML document according to the RDF Model and Syntax specification. The tool checks whether both the RDF Schemata and related Metadata instances satisfy the semantic constraints implied by the RDF Schema Specification.

The tool possesses the following features:

- Semantic and syntax checking of both RDF Schemata and Metadata instances

---

1 http://www.w3.org/RDF/Implementations/SiRPAC/
2 http://www.profium.com
3 http://www.w3.org/1999/02/26-modules/
4 http://www.ilrt.bris.ac.uk/discovery/rdf/resources/
5 http://www.w3.org/RDF/
6 http://www.ics.forth.gr/proj/isst/RDF/

- Statement validation across several RDF/XML namespaces,
- understand embedded RDF in HTML or XML, and
- fully developed in Java providing full Unicode support.

This parser makes a bridge between RDF Schemas and corresponding RDF instances and is used as a part of RDF Schema repository projects.

2.2.2 RDF Repositories

RDF repositories are built on top of RDF parsers and provide additional functions concerning storage of large amount of triples in relational databases, reformulation of RDF queries to SQL database queries, providing efficient RDF parsing and interchange functions.

The rdfDB[7] project is intended to be a simple, scalable, open-source database for RDF. The goals of this project are to build a database that is capable of

- Supporting a graph oriented API via a textual query language similar to SQL.
- Support different ways to fill the database.
- Operate in a scalable way.
- Provide support for RDF Schemas.
- Provide support for some basic forms of inference.
- Provide both C and Perl access to the database.

The rdfDB repository, which is now under intensive development, provides the following basic RDF database operations: database creation, adding triples into a database, namespace manipulation, database querying. Query interface supports only one query command, which allows retrieves a set of triples, which satisfy a certain set of constraints. The support for RDF Schemas and inference are planned, but not yet designed or implemented.

[7] http://web1.guha.com/rdfdb/

2.2.3 Sesame: an RDF Schema Repository and Querying Tool

An RDF repository is normally targeted at queries of the type 'select the triples which specify property *A* for resource *B*, having value *C* belonging to a certain value range'. An RDF Schema querying repository need to process queries of the type 'select subclasses of class *A*, which have property *B* attached to them, where *B* has a range equal to classes c and *D*'. These queries deal with the schema information, not with the triples of instance documents. An integrated RDF and RDF Schema repository may answer queries which use both schema and instance information, like 'select instances of class a, which have property B defined, and property C having value D'. These queries bring a qualitatively new functionality to RDF storage and retrieval facilities.

Sesame[8] is an RDF schema-based repository and querying service, being developed by Aidministrator[9] in the frame of the On-To-Knowledge[10] project (cf. [Fensel et al., 2000 (b)], [Ontoknowledge]). Sesame provides a substantial functionality in querying RDF instances (ontology instances) and RDF Schemas (ontologies), using an object oriented query language RQL. The ontologies and their instances are accessible in a uniform way be means of RQL queries. The RQL language implemented in Sesame allows querying class definitions and class instances by their properties, supporting an extensive set of querying expressions. An example of a query formulated to the Sesame service, and asking for all the instances of a certain property, is presented in Figure 1.

3 Ontologies

Ontologies are an essential backbone technology because the provide an important feature: they interweave formal semantics understandable by a computer with real world semantic understandable to humans. Therefore, we will take a closer look at them in this section. We will answer the question what are Ontologies, we will describe representation languages, tools, and issues related to versioning and heterogeneity.

[8] http://sesame.aidministrator.nl/
[9] http://www.aidministrator.nl
[10] http://www.ontoknowledge.org

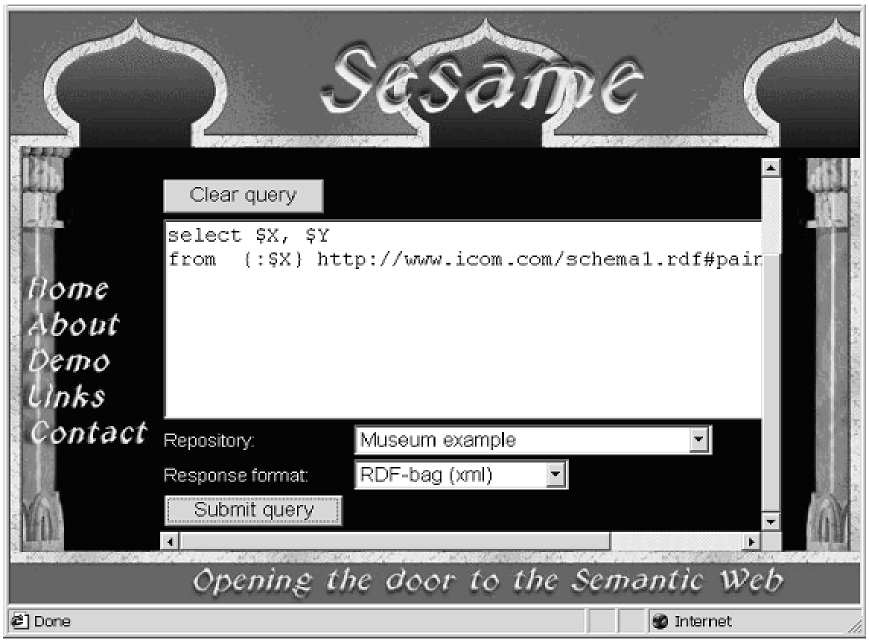

Fig. 1 Querying Sesame.

3.1 What are Ontologies

Ontologies were developed in Artificial Intelligence to facilitate knowledge sharing and reuse. Since the beginning of the nineties ontologies have become a popular research topic investigated by several Artificial Intelligence research communities, including Knowledge Engineering, natural-language processing and knowledge representation. More recently, the notion of ontology is also becoming widespread 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 in large part due to what they promise: a shared and common understanding of some domain that can be communicated between people and application systems. Because ontologies aim at consensual domain knowledge their development is often a cooperative process involving different people, possibly at different locations. People who agree to accept an ontology are said to commit themselves to that ontology.

Many definitions of ontologies have been given in the last decade, but one that, in our opinion, best characterizes the essence of an ontology is based on the related definitions by [Gruber, 1993]: An ontology is a formal, explicit specification of a shared conceptualization. A '*conceptualization*' refers to an abstract model of some phenomenon in the world which identifies the relevant concepts of that phenomenon. 'Explicit' means that the type of concepts used and the constraints on their use are explicitly defined. '*Formal*' refers to the fact that the ontology should be machine understandable. Hereby different degrees of formality are possible. Large ontologies like WordNet[11] provide a thesaurus for over 100,000 terms explained in natural language. On the other end of the spectrum is CYC[12], that provides formal axiomating theories for many aspect of common sense knowledge. '*Shared*' reflects the notion that an ontology captures consensual knowledge, that is, it is not restricted to some individual, but accepted by a group.

3.2 Languages

RDF Schema is quite simple compared to full-fledged knowledge representation languages. To be able to specify the meaning of data more precisely, richer languages are necessary. OIL[13] [Fensel et al., 2001 (a)] is such an enhanced language. OIL is the result of an initiative to integrate intuitive modeling primitives, Web-languages, and formal semantics into one language. One of the central design ideas in OIL is its onion model (see Figure 2). There will never be any one language that satisfies all human requirements. OILs onion model offers languages of varying complexity; this allows applications to select the degree of complexity they require. One of its dialects called **DAML+OIL**[14] reflects a broad European and (US) American consensus on modeling primitives for the semantic web and will be the departure point for standardization by the W3C[15]. Both OIL and DAML+OIL are defined as

---

[11] http://www.cogsci.princeton.edu/~wn
[12] http://www.cyc.com/
[13] http://www.ontoknowledge.org/oil
[14] http://www.daml.org
[15] http://www.w3c.org

an extension to RDFS, which makes them to a large extent compatible with plain RDFS [Broekstra et al., 2000]. Their main additions to RDFS are formal semantics, based on a Description Logic, and more advanced modeling primitives, such as boolean expressions and some axioms.

Further extensions to those languages are thought of. For example, Triple[16] is a recent initiative targeted development of an Open-Source Query and Inference Language and Engine for Semi-Structured Data (RDF and DAML/OIL). The language is a successor of SiLRI [Decker et al., 1998], a simple logic-based RDF interpreter. Triple is proposed as a new RDF query and inference language, providing full support for resources and their namespaces, models represented with sets of RDF triples, reification, RDF data transformation, and an expressive rule language for RDF. The language is intended to be used with any Horn-based inference engine.

3.3 Tools

Effective and efficient work with the semantic web must be supported by advanced tools enabling the full power of this technology. In particular, we need the following elements:

- Editors and semi-automatic construction to build new ontologies.

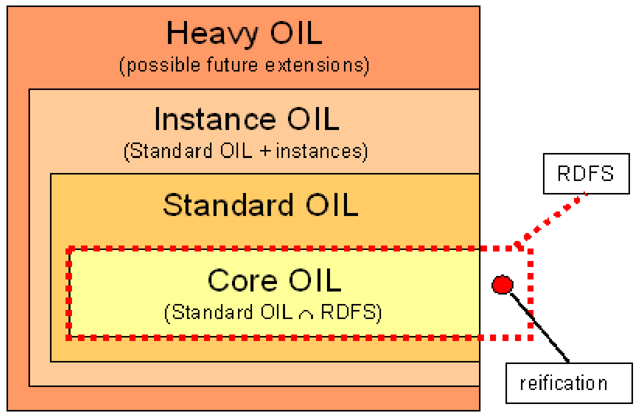

Fig. 2 The onion model to control complexity.

[16] http://www-db.stanford.edu/~stefan/2001/02/rdfinterest/

- Annotation tools to link unstructured and semi-structured information sources with meta data.
- Reasoning Service: Instance and Schema Inferences enable advanced query answering service, support ontology creation and help to map between different terminologies.
- Reusing and Merging Ontologies: Ontology library systems and Ontology Environments help to create new ontologies by reusing existing ones.

In the following, we will briefly describe examples for these technologies.

3.3.1 Editors and semi-automatic construction

Ontology editors help human knowledge engineers to build ontologies. Ontology editors support the definition of concept hierarchies, the definition attributes for concepts, and the definition of axioms and constraints. They must provide graphical interfaces and must confirm to existing standards in web-based software development. They enable inspecting, browsing, codifying and modifying ontologies and supports in this way the ontology development and maintenance task. An example system is Protégé [Grosso et al., 1999] (see Figure 3).

Manually building ontologies is a time-consuming task It is very difficult and cumbersome to manually derive ontologies from data. This appears to be true even regardless of the type of data one might consider. Natural language texts exhibit morphological, syntactic, semantic, pragmatic and conceptual constraints that interact in order to convey a particular meaning to the reader. Tools that learn ontologies from natural language exploit the interacting constraints on the various language levels (from morphology to pragmatics and background knowledge) in order to discover new concepts and stipulate relationships between concepts. Therefore, in addition to editor support, semi-automated tools in ontology development help to improve the overall productivity. An example system is Text-To-Onto (cf. [Mädche & Staab, 2000]) which provides an integrated environment for the task of learning ontologies learning from text.

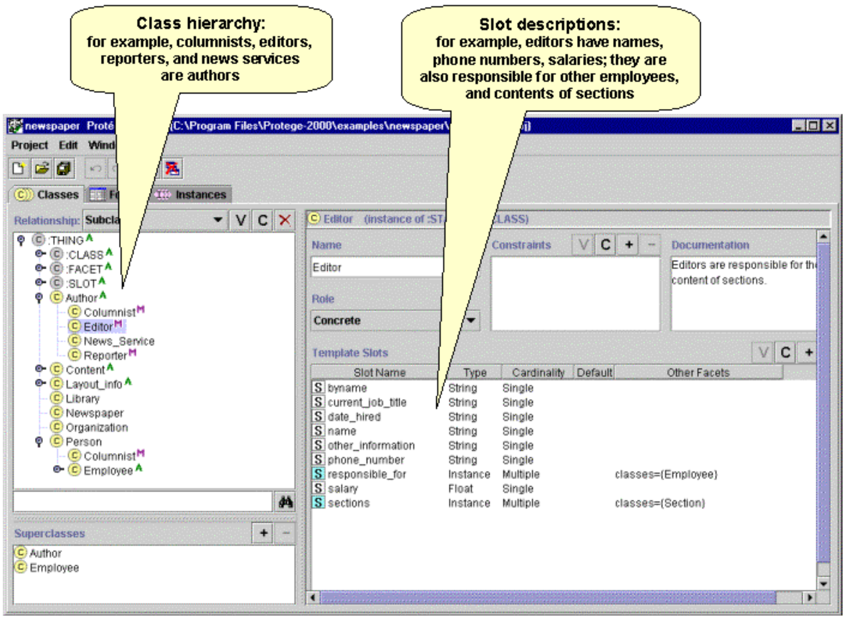

Fig. 3 The Protégé editor.

3.3.2 Annotation tools

Ontologies can be used to describe large instance population. Tools help the knowledge engineer to establish such link via: (1) Linking an ontology with a database schema or deriving a database schema from an ontology in case of structured data. (2) Deriving an XML DTD, an XML schema, and an RDF schema from an ontology in case of semi-structured data. (3) Manually or semi-automatically adding ontological annotation to unstructured data. More details can be found in [Klein et al., 2000].

3.3.3 Reasoning Service

Inference engines for ontologies can be used to reason about instances of an ontology or over ontology schemes. Reasoning over Instances of an ontology, for example, derive a certain value for an attribute applied to an object. These inference services are the equivalent of SQL query engines for databases, however provide stronger support (for example, recursive rules). An example system is Ontobroker ([Fensel et al., 2000 (a)]) which is meanwhile commercialized by the company Ontoprise[17]. Reasoning over Concepts of an ontology, for example, automatically derive the right position of a new concept in a give concept hierarchy. The FaCT (Fast Classification of Terminologies) [Horrocks & Patel-Schneider, 1999] can be used to automatically derive concept hierarchies. It is a Description Logic (DL) classifier that makes use of the well-defined semantics of OIL. Both types of reasoners help to build ontologies and to use them for advanced information access and navigation as we will discuss below.

3.3.4 Ontology Libraries and Environments

Assuming that the world is full of well-designed modular ontologies, constructing a new ontology is a matter of assembling existing ones. Instead of building ontologies from scratch one wants to reuse existing ontologies. This requires two types of tools: (1) tools to storage and retrieve existing Ontologies and (2) tools that help manipulate existing Ontologies.

How to manage Ontologies in an efficient and proper way to fortify their reuse of knowledge is far from trivial. Ontology library systems are an important environment in grouping and re-organizing ontologies for further reuse, integration, maintenance, mapping and versioning. An Ontology library system offers various functions for managing, adapting and standardizing groups of ontologies. It should be easily accessible and offer efficient support for re-using existing relevant ontologies and standardizing them based on upper-level ontologies and ontology representation languages. For this reason, an ontology library system will, at the very least, feature a functional infrastructure to store and maintain ontologies, an uncomplicated adapting environment for editing, searching and reasoning ontologies, and strong standardization support by providing upper-level ontologies and standard ontology representation languages. In a nutshell, In order to facilitate ontology reuse, a library system must support the following: (see Figure 4):

- ontology reuse by open storage, identification and versioning.
- ontology reuse by providing smooth access to existing ontologies

[17] http://www.ontoprise.de

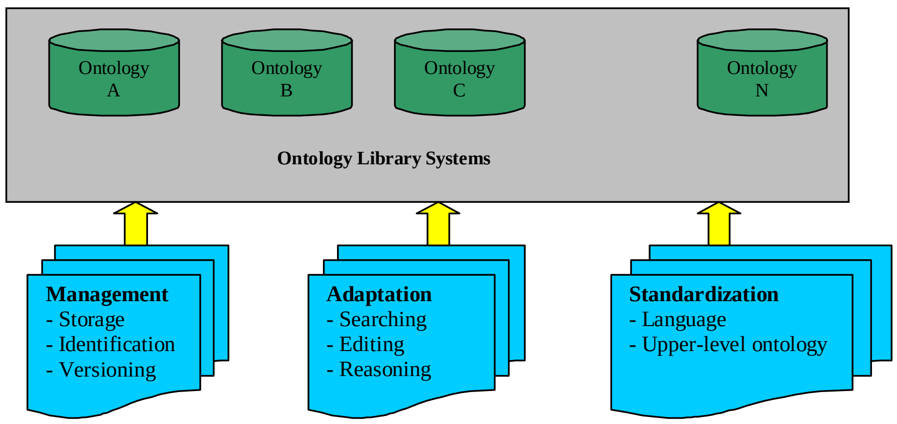

Fig. 4 Main functionality of the libary systems.

and by providing advanced support in adapting ontologies to certain domain and task-specific circumstances (instead of requiring such ontologies to be developed from scratch).

- ontology reuse by fully employing the power of standardization. Providing access to upper-layer ontologies and standard representation languages is one of the keys to developing knowledge sharing and re-use to its full potential.

Examples of Ontology library systems are: These include: WebOnto[18] [Motta, et al., 2000], Ontolingua[19] [Farquhar at al., 1997], [Duineveld et al., 1999], DAML Ontology library system[20], SHOE[21] [Heflin & Hendler, 2000], Ontology Server[22], IEEE Standard Upper Ontology [23] (IEEE), Sesame[24], OntoServer[25], and ONIONS[26]. ONIONS is a methodology for ontology integration and was successfully implemented in several medical ontology library systems [Pisanelli et al., 1998]. A

18 http://eldora.open.ac.uk:3000/webonto
19 http://www-ksl-svc.stanford.edu:5915/
20 http://www.daml.org/ontologies/
21 http://www.cs.umd.edu/projects/plus/SHOE/
22 http://www.starlab.vub.ac.be/research/dogma/OntologyServer.htm
23 http://suo.ieee.org/refs.html
24 http://sesame.aidministrator.nl/
25 http://ontoserver.aifb.uni-karlsruhe.de/
26 http://saussure.irmkant.rm.cnr.it/onto/

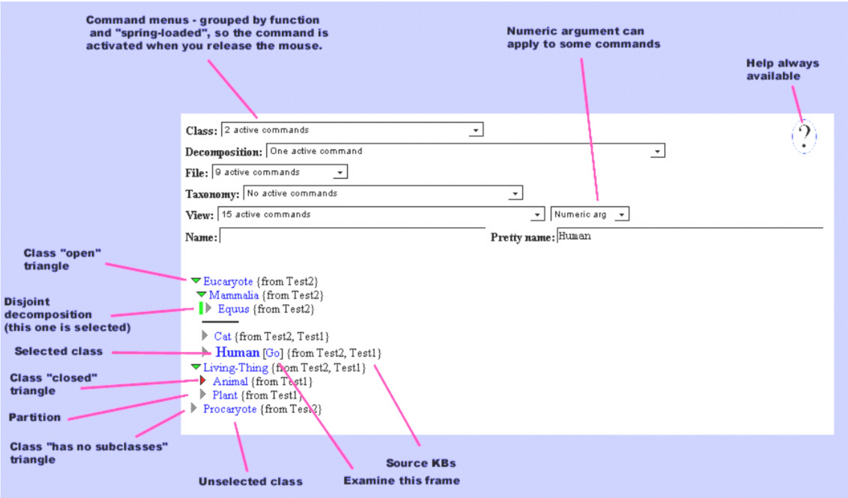

Fig. 5 Chimaera.

detailed description and comparison of these systems can be found in [Ding & Fensel, 2001].

We also require tools that provide support in adaptation and merging existing ontologies to make them fitting to new tasks and domains. Operations for combining ontologies are: Ontology inclusion, Ontology restriction, and polymorphic refinement of Ontology. E.g. inclusion of one ontology in another has the effect that the composed ontology consists of the union of the two ontologies (their classes, relations, axioms). A tool environment in this area is Chimaera which provides support for two important tasks: (1) merging multiple ontologies and (2) diagnosing (and evolving) ontologies [McGuinness et al., 2000]. It has been developed at the University of Stanford. Figure 5 illustrates Chimaera.

The PROMPT[27] tool [Noy & Musen, 2000] is available as a plug-in for Protégé-2000 targeted to help the user in ontology merging. PROMPT takes two ontologies as input and guides the user in the creation of a single merged ontology as output. First PROMPT creates an initial list of matches based on class names. Then it guides the user through the following integration cycle:

27 http://protege.stanford.edu/plugins/prompt/prompt.html

1) the user triggers an operation by either selecting one of PROMPT's suggestions from the list or by using an ontology-editing environment to specify the desired operation directly; and
2) PROMPT performs the operation, automatically executes additional changes based on the type of the operation, generates a list of suggestions for the user based on the structure of the ontology around the arguments to the last operation, and determines conflicts that the last operation introduced in the ontology and finds possible solutions for those conflicts.

The tool can be used to merge various ontologies to create the general ontology (e.g., a product catalog ontology out of standard-specific cXML, xCBL, etc. catalog formats), which is stored at the marketplace and serves as a mediator between different document formats. However, for the B2B document integration task ontology mapping (i.e. making bridges between different non-changeable ontologies) seems to be more important than ontology merging (i.e. creating a single ontology out of several other ontologies). PROMPT provides no support in this task.

3.4 Ontologies as dynamic networks of meaning

As ontologies become more popular and are also used in real-life situations, new problems arise. Two important topics that the ontology research community is currently facing are:

- **Evolving ontologies**: how to manage ontologies that change over time. Ontologies are often not stable definitions that never change. One of the reasons for this is that a *shared* conceptualization of a domain has to be reached in a process of social communication. Other reason for modification of the ontology are changes in the domain and adaptation to a different task.
  The evolution of ontologies causes operability problems, which hamper their effective reuse. Solutions are required to allow changes to ontologies without making current use invalid.
- **Combining ontologies**: how to relate and align separately developed ontologies to use them together. Nowadays, people start annotating (web)data with standard terminologies and other semantic meta data. This is providing us with a lot of freely accessible domain specific ontologies. However, to form a real *web of semantics* - which will

allow computers to combine and infer implicit knowledge - those separate ontologies should be linked and related to each other. Adaptation of existing ontologies, and composition of new ontologies from ontologies that are already around are important issues in this area.

In the remainder of this section we will discuss these two areas in more detail.

3.4.1 Ontology versioning

Coping with changing data structures is not a new topic in computer science. In the database area, there has been a lot of research on the topic of *database schema evolution*. However, although there are quite a few similarities, there are also many differences, which are discussed in detail in [Noy & Klein, to appear]. An important difference is the fact that the distinction between data and schema is not as clear as in databases. Ontologies themselves - and not just the data - are often used in applications, for example as controlled vocabularies, or navigation structures. Further, ontologies are even more distributed in nature than databases. In the case of distributed databases, there is often a clear picture of the locations where changes might have effect. In the case of ontologies used on the Semantic Web, there is completely no clue which applications are using the ontology. Synchronization or tuning of changes is not possible.

These differences make that the traditional distinctions [Roddick, 1995] between evolution (new schemas that are backward compatible) and versioning (multiple views on the data via different versions), and read and update compatibility are not very relevant for ontology versioning. Changes to ontologies *will* happen and some of them will probably cause incompatibilities. A versioning methodology for ontologies can therefore not guarantee that no information is lost, but it should make the effects of changes explicit. Ontology versioning is *the ability to manage ontology changes and their effects by creating and maintaining different variants of the ontology*. The management of changes thus is the key issue in the support for evolving ontologies [Klein & Fensel, 2001].

The mechanisms and techniques to manage those changes to ontologies should aim at achieving maximal interoperability with existing data and applications. This means that it should retain as much information and knowledge as possible, without deriving incorrect information. Such a methodology has the following elements:

- an identification mechanism: for every use of a concept or a relation, a versioning framework should provide an unambiguous reference to the intended definition;
- a change specification mechanism: the relation of one version of a concept or relation to other versions of that construct should be made explicit, both by specifying the ontological relation (e.g. subclass of) and the intention of the change (e.g. replacement).
- transparent access: methods to give a valid interpretation to as much data as possible, i.e. automatically translate and relate the versions and data sources as far as possible.

Altogether, we see that ontology versioning is even more complicated than database versioning. However, it is an absolutely necessary element of the Semantic Web. After all, if it is not clear which version of the meta-data describes which version of the data, the reasoning about the data cannot be trusted any more.

3.4.2 Aligning and relating

To achieve the full promise of the Semantic Web, meta-data and knowledge from different sources should be combined to reason about data and perform the automated data processing. Combining independently developed ontologies is therefore a central aspect of the Semantic Web. This is, however, far from trivial and requires considerable effort [Uschold et al., 1998]. Ontologies may differ at many dimensions, ranging from terminological and ontology-language differences to paradigm differences [Klein, 2001]. To combine ontologies, many of those differences must be solved.

There are several ways to actually combine ontologies, sometimes summarized under the term “ontology integration” [Pinto et al., 1999]. We make the following distinction, based on the extent of integration of the original ontologies.

- **merging**: creating one new ontology from two or more ontologies. In this case, the new ontology will unify and replace the originating ontologies. This often requires considerable adaptation and extension.

- **aligning**: bringing the ontologies into mutual agreement. In this case, the ontologies are kept separate, but at least one of the original ontologies is adapted in such a way that the conceptualization and the vocabulary match in the overlapping parts of the ontologies. However, the ontologies might describe different parts of the domain in different levels of detail.
- **relating**: specifying how the concepts in the different ontologies are related in a logical sense. This means that the original ontologies are not changed, but that additional axioms describe the relation between the concepts. Leaving the original ontologies unchanged often implies that only a part of the integration can be done, because major differences may require adaptation of the ontologies.

Notice that merging normally involves aligning as a first step. There are several tools available for this task, based on different types of heuristics [Klein, 2001]. A few of these tools had been described in the section on ontology tools.

The choice for merging, aligning or relating is often driven by the type of application. Applications with a central control tend to further integration than distributed applications. Therefore, ontology combination on the Semantic Web will probably heavily rely on techniques for *relating* ontologies. Proven methodologies and techniques for relating and adapting ontologies are thus necessary.

4 Application Areas

In the following we will discuss two promising application areas for semantic web and ontology technology. The first one is concerned with information access as basis for effective and efficient knowledge management. The second one is about optimizing business relationships based on intelligent Ebusiness.

4.1 Knowledge Management

Knowledge Management is concerned with acquiring, maintaining, and accessing knowledge of an organization. It aims to exploit an organizations intellectual assets for greater productivity, new value, and increased competitiveness. Due to globalization and the impact of the Internet, many organizations are increasingly geographically dispersed and organized around virtual teams. With the large number of on-line documents, several document management systems entered the market. However these systems have severe weaknesses:

- **Searching information**: Existing keyword-based search retrieves irrelevant information which uses a certain word in a different context, or it may miss information where different words about the desired content are used.
- **Extracting information**: Human browsing and reading is currently required to extract relevant information from information sources, as automatic agents lack all common sense knowledge required to extract such information from textual representations, and they fail to integrate information spread over different sources.
- **Maintaining weakly structured text sources**: It is a difficult and time-consuming activity when such sources become large. Keeping such collections consistent, correct, and up-to-date requires a mechanized representation of semantics and constraints that help to detect anomalies.
- **Automatic document generation**: Adaptive web sites which enable a dynamic re-configuration according to user profiles or other relevant aspects would be very useful. The generation of semi-structured information presentations from semi-structured data requires a machine-accessible representation of the semantics of these information sources.

Using Ontologies, semantic annotations will allow structural and semantic definitions of documents providing completely new possibilities: Intelligent search instead of keyword matching, query answering instead of information retrieval, document exchange between departments via ontology mappings, and definition of views on documents.

Ontobroker (cf. [Fensel et al., 2000 (a)]) applies Artificial Intelligence techniques to improve access to heterogeneous, scattered and semi-structured information sources as they are presented in the World Wide Web or organization-wide intranets. It relies on the use of *ontologies* to annotate web pages, formulate queries, and derive answers. The gist of the matter is: to define an ontology and use it to annotate/structure/wrap your web documents, and somebody else can make use of Ontobroker's advanced query and inference services to consult your knowledge. To achieve this goal, Ontobroker provides three interleaved languages and two tools. It provides a broker architecture with three core elements: a query interface for formulating queries, an inference engine used to derive answers, and a webcrawler used to collect the required knowledge from the Web. It provides a *representation* language for formulating ontologies. A subset of it is used to formulate queries, i.e. to define the *query language*. An *annotation* language is offered to enable knowledge providers to enrich web documents with ontological information. The strength of Ontobroker is the close coupling of informal, semiformal, and formal information and knowledge. This supports their maintenance and provides a service that can be used more generally for integrating knowledge-based reasoning with semi-formal represented documents.

The concepts and techniques used by Ontobroker have been further developed in the Ontoknowledge project (cf. [Ontoknowledge]). The Ontoknowledge architecture and all its major components are shown in Figure 6. To illustrate these components and their interactions, we present a simple querying scenario, where a user poses a query to the system that must be answered on the basis of a set of weakly structured data sources in a repository. The sequence of numbers in Figure 1 indicates the steps that must be taken in order to perform any of the above queries.

- Step [1]. The system interacts with a user in order to elicit a specific query to be answered. Both the interaction with the user and the resulting query are entirely in terms of a domain-specific ontology, expressed in the OIL language developed within the consortium (see Section 2 and 3). The required ontologies are constructed using tools such as OntoEdit developed by the University of Karlsruhe. Such an ontology-based user interaction has as main advantage that the user is shielded from any document-specific representation details, and can instead communicate in meaningful domain-specific terms. Furthermore, it makes the system much more robust against changes and variability in the formats of the underlying documents.
- Step [2]. The user interaction results in a specific query to be answered by the data repository layer. We rely on the Resource

Description Framework currently being developed by the World-Wide Web consortium (W3C), to structure the data repository and to express queries over this repository. The required translation from OIL-based user interaction to RDF-based queries is feasible because OIL is itself definable in terms of RDF-Schema definitions. Searching the full text of documents along with any associated RDF annotations can ensure the high recall desirable in early stages of the retrieval process. To accommodate this information seeking behavior, RD*Fferret* combines full text searching with RDF querying. RD*Fferret* closely cooperates with BT's full text search engine WebFerret. The user can use RD*Fferret* like a conventional Internet search engine by entering and refining a set of search terms or by entering a natural language query.

- Step [3]. The consortium has developed an RDF Schema query engine to efficiently process queries over medium-size data-repositories (with up to a million RDF triples in the repository). Sesame is an RDF Schema-based Repository and Querying facility.

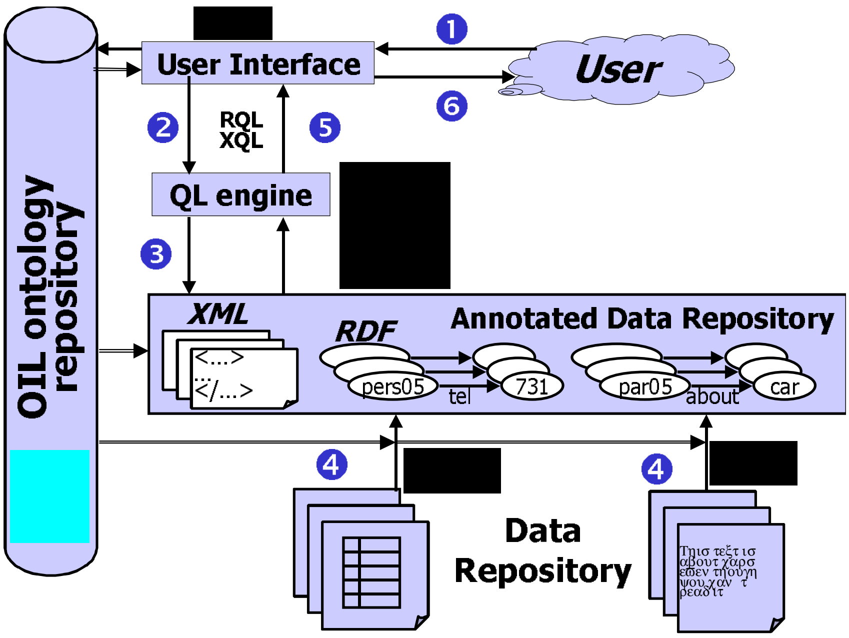

Fig. 6 The Ontoknowledge toolset for ontology-based knowledge management.

It is being developed by Aidministrator. Sesame supports highly expressive querying of RDF data and schema information, using an OQL-style query language, called RQL, which is being developed by the ICS-FORTH institute in Greece. Sesame supports storing of large quantities of RDF and RDF Schema information. The RDF is parsed using the SiRPAC parser, and stored in the Object-Relational DBMS PostgreSQL.

- Step [4]. Of course, the above steps all assume that the data repository is filled with data that is annotated with sufficiently rich semantic information. Furthermore, the annotations must be related to the ontological vocabulary that was the basis for the original user query. Different technologies will be exploited to provide these annotations, depending on whether we are dealing with weakly structured data sources, or data sources that consist of free text only. In the first case, we will use wrapper technology. In the second case, the Corporum technology from CognIT (cf. [Bremdal et al., 1999]) is the main platform for concept extraction from free text. Other tools are based on automated summarization technology as developed for ProSum by BT (cf. [Davies et al., 1998]).
- Steps [5,6]. After the RDF query has been executed over the data repository, the resulting information is communicated to the user. Again, this must be done using an ontology-based vocabulary. Furthermore, powerful graphical visualizations of query results in the context of large data sets are developed. Examples of such visualizations are the semantic sitemaps produced by the WebMaster tool of AIdministrator (see Figure 7 and [van Harmelen & van der Meer, 1999]).

On-To-Knowledge is carrying out three industrial case studies to evaluate the tool environment for ontology-based knowledge management and the associated web inference layer OIL. These case studies are chosen such that they ensure a broad coverage, involving three different industry sectors (insurance, telecom, energy) in three different countries, and facing different knowledge management problems.

In addition to the tool set and the OIL language, On-To-Knowledge is developing an associated methodology for ontology-based knowledge management. Input to this are existing European research results, such as the CommonKADS approach to knowledge engineering and management [Schreiber et al., 2000], experiences from knowledge-based software engineering and tool development, ontology composition and information retrieval techniques, and feedback from the industrial case studies.

Meanwhile new projects continue this track of research. [H-Techsight] applies semantic web technology to knowledge management solutions for technology-intensive areas.

4.2 E-commerce

Nowadays e-business includes thousands of companies, which provide a tremendous number of products for electronic markets. Forecasts for the dollar value of Business-to-Business (B2B) e-commerce in US range between $634 billion and $3.9 trillion. Already approximately 3 in 5 companies are using e-commerce to some extent and a further 20 percent

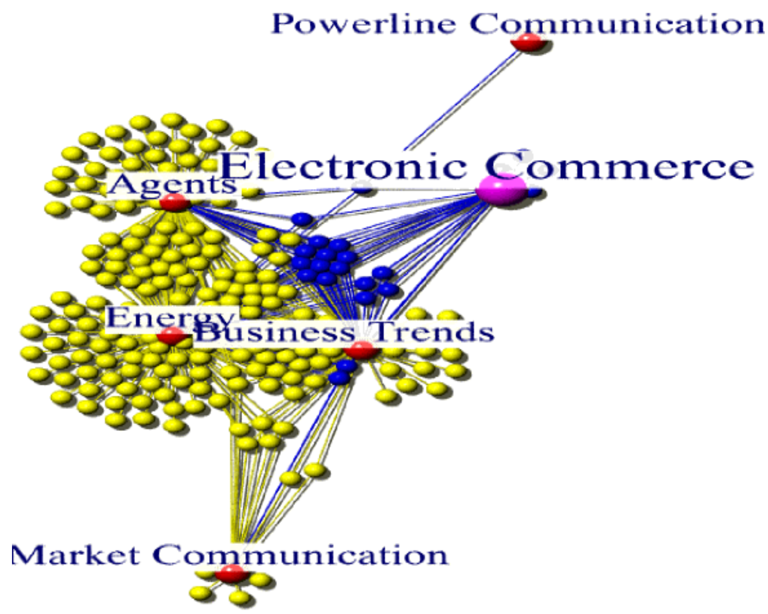

Fig. 7 Automatically generated semantic structure maps of a website.

say they intend to do so in the future. By 2003 some 80 percent of all business-to-business transactions could occur on-line [US Department of Commerce, 2001]. Unlike traditional markets, the electronic markets allow the participants to easily compare the offers and perform business negotiations faster due to electronic information interchange provided to market participants by B2B marketplaces. B2B marketplaces bring together up to tens thousands on-line suppliers and buyers each, and mediate their negotiation and purchase transactions. Different studies estimate up to 10,000 B2B marketplaces to appear in the very near future. However, a simplified view on the mission and required functionality of the marketplaces has lead to recent bankrupts and general disappointment in the B2B mediation as a whole.

To be successful and to impact the businesses, B2B marketplaces need to possess the following functionality:

- Translate between various document representations used in B2B procurement.
- Mediate different product and service ontologies, which provide formal description and hierarchies of products and services.
- Mediate different e-business ontologies, which specify the business processes, the roles of market participants, and value transformation chains.

Each of these tasks is crucially important for e-business, and might be solved by applying Semantic Web technologies.

4.2.1 B2B Document Transformation

Most of the B2B participants encode their documents in XML, however, a large number of XML serializations for conceptually equivalent documents have been developed until now. They differ in the document structures and information decomposition, complexity, and usage scenarios. Traditional and ad-hoc transformation technologies have appeared not sufficient to perform necessary transformations in an efficient and scalable way (cf. [Omelayenko & Fensel, 2001a] and [Omelayenko & Fensel, 2001b] for a relevant discussion). The solution can be sought in the ontology technology, one of the most promising technologies for e-business [Fensel, 2001]. Ontology-mediated information integration provides an elegant framework for document integration, bringing conceptual structure to the documents, and allowing further integration with knowledge-based document search and retrieval.

The B2B document transformation tasks are quite similar to those to be solved while building the Semantic Web. They are:

- Constructing RDF conceptual models from XML documents, which is known in the Semantic Web as ontology-based data annotation.
- Aligning different document ontologies.
- Performing ontology-based information querying and retrieval to store an align document ontologies.
- Performing complicated and knowledge-intensive information transformation, including generation of conceptually different views on the same knowledge.

These Semantic Web technologies are needed to resolve the B2B document integration problems, and the latter may provide the guidelines for the Semantic Web development.

4.2.2 Product Ontologies

Content standards provide hierarchies of product descriptions and define the subclass-of relationship between product categories to be used for intelligent product search, product composition and cost allocation. Each product from a product catalog has an attached link to a certain product category, which actually describes the product.

The most well-known content standard UNSPSC[28] has a 5-level classification scheme with more than 12,000 categories. It is not descriptive, that is, it contains no attributes for products but only the hierarchy of product names. The hierarchy support the suppliers' view on the classification: printing and writing paper appear together under the 'Printing Products' family. However, pens belong to the 'Office Equipment' category, because they belong to a completely different group of products from the suppliers' view.

Another content standard, ecl@ss[29], supports the flow of products and

[28] http://www.unspsc.org
[29] http://www.eclass.de

information along the supply chain of an industrial enterprise and it is mainly used in Germany. It provides more than 12,700 categories with attributes tailored to the needs of industrial customers and their suppliers. The hierarchy is build according to the needs of buyers, and, for example, ‘Writing paper’ belongs to ‘Office supplies’ together with ‘Pens’.

Essentially, content standards are product ontologies, that provide the hierarchy of product classes, together with the associated product attributes. Accordingly, the content management tasks needed to be solved by the marketplaces are essentially a ontology management tasks, which also appear in the Semantic Web. The marketplaces need to solve the following (product) ontology management tasks (cf. [Fensel et al., 2001 (a)] for a detailed discussion):

- Product *classification*: when a marketplace must assign a certain class from the product ontology to a product, given a textual product description as input.
- Product *re-classification*: when a marketplace must assign a new class to a product according to a new product ontology, given the product classification in some other product ontology. This requires application of non-trivial ontology aligning techniques.
- *Personalization* of product hierarchies and view generation on different product ontologies.
- *Maintaining* different versions of the ontologies, required because of frequent changes in the content standards (e.g. updates to UNSPSC appear every two weeks).

Hence, management of product ontologies (also known as content management) serves as a prominent application of the Semantic Web technologies, which provides a lot of practical experiences and requirements for the latter.

4.2.3 E-Business Ontologies

E-business ontologies are currently in the process of being developed and accepted by the community. The first generation of them is primary dealing with formalizing business processes, focusing on the technical side.

The TOVE[30] project resulted at several ontologies, which specify various aspects of an enterprise. Modeling of an enterprise was guided by different sets of constraints on the processes to be executed inside an enterprise. The Process Interchange Format PIF[31] aims at the development of a unified language to share and interweave heterogeneous business process models across different formats and schemas.

The ebXML[32] initiative provides a set of specifications that enable a modular electronic business framework. The vision of ebXML is to enable a global electronic marketplace where enterprises of any size and in any geographical location can meet and conduct business with each other through the exchange of XML-based messages.

The second generation of e-business ontologies focuses on the development of new models of doing web-powered business (cf. [Gordijn et al., 2000]). The OBELIX [Obelix] project is the first e-business ontology project of the second generation. It focuses on providing smart scalable integration and interoperability capabilities needed for dynamic value constellations in e-business, which deal with quite complex products and services, supply chains and value networks.

5 Conclusions

The paper provides a survey on a newly arising research are called *Semantic Web*. Semantic web technology should bring access to on-line available information to a new and much higher level of service. This higher level of service overcomes serious limitations of current web technology in finding, extracting, interpreting, and processing information. Currently most of these tasks are left to the human user. Semantic web technology will provide a highler level of mechanization for many of these information processing processes. This service is based on machine-processable semantics of data enabling information processing via a computer. We discussed the main pieces that are required to build the “data infrastructure” of the semantic web. We need languages for representing meta data, we need Ontologies that link formal with real world semantics, and we need various tools and convincing application areas to keep the ball rolling.

---

[30] http:// www.eil.utoronto.ca/tove/toveont.html
[31] http://www.eil.utoronto.ca/PIF/pif.html
[32] http://www.ebxml.org

Semantic web technology is still in its early stage. We are focussing on building its basic and mostly static infrastructure. The next step will be to realize active components on top that make use of this infrastructure to provide intelligent services to human user. Web services aim on supporting information access and Ebusiness. Examples are UDDI[33], a repository for describing vendors, products, and services. It uses WSDL[34] to describe its entries and SOAP[35] as a protocol that defines how they can be accessed. Currently all these service description elements are not yet based on semantic web technology. Therefore, searching for vendors, products, and services; comparing and combing products; coalitation forming of vendors etc. require severe human effort. A highler level of service based on mechanizing many of these aspects can be provided via semantic web enabled services. Steps in this directions are taken by projects such as DAML[36] and Ibrow [Ibrow]. Within DAML, a service description language called DAML-S [Ankolenk et al., 2001] has been developed. It allows formal competence descriptions that enable automatic inference as a means to select and combine services. Ibrow developed a language called UPML [Fensel et al., to appear] that can be used to describe static and dynamic aspects of a semantic web. It provides elements to describe Ontologies, heuristic reasoners (called problem-solving methods) and means to interweave them. Based on these description an automated broker provide support in component selection, combination, and execution.

The semantic web is still mainly a vision. First steps into its direction have been taken. There are first languages, tools, and application done. Still most of the way is still ahead.

References

[Ankolenk et al., 2001]
A. Ankolenkar, M. Burstein, J. R. Hobbs, O. Lassila, D. L. Martin, S. A. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, H. Zeng: DAML-S: A Semantic Markup Language For Web Services. In *Proceedings of the Semantic Web Working Symposium (SWWS)*, Stanford, California, May 2001.

33 http://www.uddi.org

34 http://www.wsdl.org

35 http://www.soap.org

36 http://www.daml.org

[Berners-Lee et al., 2001]
T. Berners-Lee, J. Handler, and O. Lassila: The Semantic Web, *Scientific American*, May 2001.

[Bremdal et al., 1999]
B. Bremdal, F. Johansen, C. Spaggiari, R. Engels, and R. Jones: Creating a Learning Organisation Through Content-Based Document Management, OECD HRP Meeting, Loen, NO. CognIT Report, Oslo, May 1999.

[Broekstra et al., 2000]
J. Broekstra, M. Klein, S. Decker, D. Fensel, F. van Harmelen, and I. Horrocks: Enabling knowledge representation on the Web by Extending RDF Schema. In *Proceedings of the Tenth International World Wide Web Conference (WWW10)*, Hong Kong, May 2001.

[Davies et al., 1998]
J. Davies, S. Stewart, and R. Weeks: Knowledge Sharing over the World Wide Web, *WebNet '98*, Florida, USA, November 1998.

[Decker et al., 1998]
S. Decker, D. Brickley, J. Saarela, and J. Angele: A Query and Inference Service for RDF. In *Proceedings of the W3C Query Languages Workshop (QL-1998)*, December 5, 1998. [http://www.w3.org/TandS/QL/QL98/pp/queryservice.html]

[Ding & Fensel, 2001] Y. Ding and D. Fensel:
Ontology Library Systems: The key for successful Ontology Reuse. In *Proceedings of the the 1st Semantic web working symposium*, Stanford, CA, USA, July 30th-August 1st, 2001.

[Duineveld et al., 1999]
A. J. Duineveld, R. Stoter, M. R. Weiden, B. Kenepa, V. R. Benjamins: WonderTools? A comparative study of ontological engineering tools. In *Proceedings of the 12th International Workshop on Knowledge Acquisition, Modeling and Management (KAW'99)*, Banff, Canada, October, 1999.

[Farquhar at al., 1997]
A. Farquhar, R. Fikes, & J. Rice: The Ontolingua server: Tools for collaborative ontology construction. *International Journal of Human Computer Studies* 46: 707-728, 1997.

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

[Fensel et al., 2000 (a)]
D. Fensel, S. Decker, M. Erdmann, H.-P. Schnurr, R. Studer, and A. Witt: Lessons learned from Applying AI to the Web, *Journal of Cooperative Information Systems*, 9(4), 2000.

[Fensel et al., 2000 (b)] D. Fensel, F. van Harmelen, H. Akkermans, M. Klein, J. Broekstra, C. Fluyt, J. van der Meer, H.-P. Schnurr, R. Studer, J. Davies, J. Hughes, U. Krohn, R. Engels, B. Bremdahl, F. Ygge, U. Reimer, and I. Horrocks: OnToKnowledge: Ontology-based Tools for Knowledge Management. In *Proceedings of the eBusiness and eWork 2000 (EMMSEC 2000) Conference*, Madrid, Spain, October 2000.

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

[Fensel et al., 2001 (b)]
D. Fensel, Y. Ding, B. Omelayenko, E. Schulten, G. Botquin, M. Brown, and A. Flett, Product Data Integration for B2B E-Commerce, *IEEE Intelligent Systems*, vol. 16, no. 4, 2001, pp. 54-59.

[Fensel et al., to appear]
D. Fensel, E. Motta, F. van Harmelen, V. R. Benjamins, M. Crubezy, S. Decker, M. Gaspari, R. Groenboom, W. Grosso, M. Musen, E. Plaza, G. Schreiber, R. Studer, and B. Wielinga: The Unified Problem-solving Method Development Language UPML, to appear in *Knowledge and Information Systems (KAIS): An International Journal.*

[Fensel & Musen, 2001]
D. Fensel and M. Musen: Special Issue on Semantic Web Technology, *IEEE Intelligent Systems (IEEE IS)*, 16(2), 2001.

[Gordijn et al., 2000]
J. Gordijn, H. Akkermans and J. van Vliet, Business Modelling Is Not Process Modelling, In: *Conceptual Modeling for E-Business and the Web (ECOMO-2000)*, Springer-Verlag, LNCS 1921, Salt Lake City, USA, October 9-12, 2000, pp. 40-51.

[Grosso et al., 1999]
W. E. Grosso, H. Eriksson, R. W. Fergerson, J. H. Gennari, S. W. Tu, and M. A. Musen. Knowledge Modeling at the Millennium (The Design and Evolution of Protege-2000). In the *Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management (KAW-1999),* Banff, Alberta, Canada, October 16-21, 1999.

[Gruber, 1993]
T. R. Gruber: A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition*, 5:199—220, 1993.

[van Harmelen & van der Meer, 1999]
F. van Harmelen and J. van der Meer: WebMaster: Knowledge-based Verification of Web-pages. In *Proceedings of the 2nd International Conference on The Practical Applications of Knowledge Management (PAKeM99)*, London, UK, April 1999, pp. 147-166. The Practical Applications Company, Blackpool, UK, 1999.

[Heflin & Hendler, 2000]
J. Heflin and J. Hendler: Dynmaic ontologies on the Web. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*. AAAI/MIT Press, Menlo Park, CA, 2000. pp. 443-449.

[Hjelm, 2001]
J. Hjelm: *Creating the Semantic Web with RDF*, Willey Computer Publishing, 2001.

[H-Techsight]
Project in the Information Society Technologies (IST) Programme for Research, Technology Development & Demonstration under the 5th Framework Programme: *H-TECHSIGHT: A knowledge management platform with Intelligence & Insight capabilities for Technology Intensive Industries*.

[Horrocks & Patel-Schneider, 1999]
I. Horrocks and P. F. Patel-Schneider: Optimizing description logic subsumption, *Journal of Logic and Computatio*n, 9(3):267–293, 1999.

[Ibrow]
Project in the Information Society Technologies (IST) Programme for Research, Technology Development & Demonstration under the 5th Framework Programme: *IBROW: An Intelligent Brokering Service for*

Knowledge-Component Reuse on the World-Wide Web. http:// www.swi.psy.uva.nl/projects/ibrow/home.html

[Klein, 2001] M. Klein: Combining and relating ontologies: an analysis of problems and solutions. In A. Gómez-Pérez, M. Gruninger, H. Stuckenschmidt, and M. Uschold, editors, *Workshop on Ontologies and Information Sharing, IJCAI'01*, Seattle, USA, 2001.

[Klein & Fensel, 2001]
M. Klein and D. Fensel: Ontology versioning on the Semantic Web, In *Proceedings of the the 1st Semantic Web Working Symposium*, Stanford, CA, USA, July 30th-August 1st, 2001.

[Klein et al., 2000]
M. Klein, D. Fensel, F. van Harmelen, and I. Horrocks: The relation between ontologies and XML schemas, to appear in *Electronic Transactions on Artificial Intelligence (ETAI).*

[Mädche & Staab, 2000]
A. Mädche and S. Staab: Mining Ontologies from Text. In *Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference (EKAW-2000)*, R. Dieng et al. (eds.), Lecture Notes in Artificial Intelligence, LNAI, Springer-Verlag, October 2000.

[McGuinness et al., 2000]
D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder: The Chimaera Ontology Environment. In the *Proceedings of the The Seventeenth National Conference on Artificial Intelligence (AAAI 2000)*, Austin, Texas, July 30 - August 3, 2000.

[Motta, et al., 2000]
E. Motta, S. Buckingham-Shum, and J. Domingue: Ontology-Driven Document Enrichment: Principles, Tools and Applications. *International Journal of Human-Computer Studies*, 52:1071-1109, 2000.

[Noy & Klein, to appear]
N. Noy and M. Klein: Ontology Evolution: Not the Same as Schema Evolution, to appear.

[Noy & Musen, 2000]
N. Noy and M. Musen: PROMPT: Algorithm and Tool for Automated

Ontology Merging and Alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, Austin, Texas, July 30 - August 3, 2000.

[Obelix]
Project in the Information Society Technologies (IST) Programme for Research, Technology Development & Demonstration under the 5th Framework Programme: *OBELIX: Ontology-Based ELectronic Integration of compleX products and value chains*.

[Omelayenko & Fensel, 2001a]
B. Omelayenko and D. Fensel: An Analysis of Integration Problems of XML-Based Catalogs for B2B Electronic Commerce, In: *Proceedings of the Ninth IFIP 2.6 Working Conference on Database Semantics*, Hong Kong, April 25-28, 2001, pp. 232-246.

[Omelayenko & Fensel, 2001b]
B. Omelayenko and D. Fensel: A Two-Layered Integration Approach for Product Information in B2B E-commerce, In: *Proceedings of the Second International Conference on Electronic Commerce and Web Technologies (EC WEB-2001)*, LNCS 2115, Munich, Germany, September 4-6, 2001, pp. 226-239.

[Ontoknowledge]
Project in the Information Society Technologies (IST) Programme for Research, Technology Development & Demonstration under the 5th Framework Programme: *On-To-Knowledge: Content-driven Knowledge-Management Tools through Evolving Ontologies*. http://www.ontoknowledge.com.

[Pinto et al., 1999] H.S. Pinto, A. Gómez-Pérez, and J.P. Martins: Some issues on ontology integration. In *Proceedings of the Workshop on Ontologies and Problem Solving Methods during IJCAI-9*9, Stockholm, Sweden, 1999.

[Pisanelli et al., 1998]
D. M. Pisanelli, A. Gangemi, and G. Steve: An Ontological Analysis of the UMLS Metathesaurus, *Journal of American Medical Informatics Association* 5:810-814, 1998.

[Roddick, 1995] J. F. Roddick: A survey of schema versioning issues for database systems. *Information and Software Technolog*y, 37(7):383–393, 1995.

[Schreiber et al., 2000]
A.Th. Schreiber, J.M. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van De Velde, and B. Wielinga: *Knowledge Engineering and Management*, The MIT Press, Cambridge, MA, 2000.

[Uschold et al., 1998] M. Uschold, M. Healy, K. Williamson, P. Clark, and S. Woods: Ontology reuse and application. In N. Guarino, editor, *Formal Ontology in Information Systems (FOIS'98)*, Treno, Italy, June 6-8, 1998. IOS Press, Amsterdam.

[US Department of Commerce, 2001]
US Department of Commerce: Leadership for the New Millennium, Delivering on Digital Progress and Prosperity, *Third annual report the Electronic Commerce Working Group at the US Department of Commerce*, 16 January, 2001.